CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Desarollo de un servicio de enlace entre HL7 RIM y terminologías médicas

Autor:Santiago Aso Lete

Tutor:David Pérez del Rey

MADRID, ABRIL DE 2013

Índice general

Capítulo 1

RESUMEN

Hoy día, en la era post genómica, los ensayos clínicos de cáncer implican la colaboración de diversas instituciones. El análisis multicéntrico y retrospectivo requiere de métodos avanzados para garantizar la interoperabilidad semántica. En éste escenario, el objetivo de los proyectos EURECA e INTEGRATE es proporcionar una infraestructura para compartir conocimientos y datos de los ensayos clínicos post genómicos de cáncer.

Debido en gran parte a la gran complejidad de los procesos colaborativos de las instituciones, provoca que la gestión de una información tan heterogénea sea un desafío dentro del área médica. Las tecnologías semánticas y las investigaciones relacionadas están centradas en búsqueda de conocimiento de la información extraída, permitiendo una mayor flexibilidad y usabilidad de los datos extraídos. Debido a la falta de estándares adoptados por estas entidades y la complejidad de los datos procedentes de ensayos clínicos, una capacidad semántica es esencial para asegurar la integración homogénea de ésta información. De otra manera, los usuarios finales necesitarían conocer cada modelo y cada formato de dato de las instituciones participantes en cada estudio.

Para proveer de una capa de interoperabilidad semántica, el primer paso es proponer un"Common Data Model" (CDM) que represente la información a almacenar, y un "Core Dataset" que permita el uso de múltiples terminologías como vocabulario compartido. Una vez que el "Core Dataset" y el CDM han sido seleccionados, la manera en la que realizar el mapping para unir los conceptos de una terminología dada al CDM, requiere de una mecanismo especial para realizar dicha labor. Dicho mecanismo, debe definir que conceptos de diferentes vocabularios pueden ser almacenados en determinados campos del modelo de datos, con la finalidad de crear una representación común de la información.

El presente proyecto fin de grado, presenta el desarrollo de un servicio que implementa dicho mecanismo para vincular elementos de las terminologías médicas

SNOMED CT, LOINC y HGNC, con objetos del "Health Level 7 Reference Information Model" (HL7 RIM). El servicio propuesto, y nombrado como TermBinding, sigue las recomendaciones del proyecto TermInfo del grupo HL7, pero también se tienen en cuenta cuestiones importantes que surgen al enlazar entre las citadas terminologías y el modelo de datos planteado.

En éste proceso de desarrollo de la interoperabilidad semántica en ensayos clínicos de cáncer, los datos de fuentes heterogéneas tienen que ser integrados, y es requisito que se deba habilitar una interfaz de acceso homogéneo a toda ésta información. Para poder hacer unificar los datos provenientes de diferentes aplicaciones y bases de datos, es esencial representar todos éstos datos de una manera canónica o normalizada. La estandarización de un determinado concepto de SNOMED CT, simplifica las recomendaciones del proyecto TermInfo del grupo HL7, utilizadas para poder almacenar cada concepto en el modelo de datos. Siguiendo éste enfoque, la interoperabilidad semántica es conseguida con éxito para conceptos SNOMED CT, sean o no post o pre coordinados, así como para las terminologías LOINC y HGNC. Los conceptos son estandarizados en una forma normal que puede ser usada para unir los datos al "Common Data Model" basado en el RIM de HL7.

Aunque existen limitaciones debido a la gran heterogeneidad de los datos a integrar, un primer prototipo del servicio propuesto se está utilizando con éxito en el contexto de los proyectos EURECA e INTEGRATE. Una mejora en la interoperabilidad semántica de los datos de ensayos clínicos de cáncer tiene como objetivo mejorar las prácticas en oncología.

Capítulo 2

INTRODUCCIÓN

2.1. Planteamiento del problema

El presente proyecto fin de grado se encuentra ubicado dentro de dos proyectos de investigación de ámbito europeo, INTEGRATE y EURECA. El objetivo de ambos proyectos es permitir una integración completa, segura, escalable y consistente de los datos provenientes de historiales médicos electrónicos (EHR) que son utilizados en ensayos clínicos en oncología. De ésta manera se facilitará el acceso, la compartición y el intercambio de datos y conocimientos entre investigadores médicos, permitiéndoles el trabajo con una mayor diversidad de fuentes de información sobre las que desarrollar sus estudios, siendo capaces de construir modelos predictivos, identificar marcadores biológicos y resolver cuestiones durante su investigación de manera más rápida y segura.

Conseguir integrar toda ésta información permite además proveer de una de las herramientas con la que dinamizar la fase de elegibilidad de pacientes para la realización de ensayos clínicos en oncología. Ésta fase tiene por objetivo el establecer unos criterios (edad, tipo y estadio del cáncer, clases de tratamiento de cáncer recibido, etc) por los que decidir si determinado paciente debe o no participar en determinado ensayo, o para que ensayos puede ser objeto determinado paciente. Integrar la información médica de los pacientes y gestionar una serie de criterios de exclusión/inclusión a través de un razonamiento semántico facilitará los procesos de selección de pacientes para los ensayos clínicos.

La infraestructura que se está desarrollando actualmente, que posibilita ésta mejora en las prácticas de los investigadores médicos, tiene como parte fundamental una capa de interoperabilidad semántica entre terminologías médicas. En las que se expresa la información contenida en los historiales electrónicos médicos y otros

medios utilizados actualmente para contener todo tipo de información clínica, y un modelo de datos en el que poder almacenar y estructurar ésta información. Uno de los problemas que se plantea, es la gran expresividad que tienen algunas terminologías médicas (SNOMED CT, LOING, HGNC, etc) para expresar información. Ésta expresividad implica que la información contenida por determinado término depende de la estructura y el vocabulario expresado, siendo ésta cuestión uno de los principales desafíos del proyecto.

Los componentes básicos en los que sintetizar la infraestructura que se está desarrollando en estos proyectos son: crear un modelo de información común (Common Information Model) con el que poder homogeneizar toda la diversidad de información de diferentes ensayos clínicos; sustentado en un modelo común de datos (Common Data Model) en el que poder almacenar toda la información clínica; y un "Core Dataset", que contiene las ontologías de las terminologías médicas usadas, así como todo el razonamiento semántico sobre ellas para poder unir elementos del modelo común de información y el modelo común de datos.

El "Common Data Model" (CDM) está basado en el modelo de referencia de la información desarrollado por el grupo "Health Level 7" (HL7 RIM). El objetivo de éste modelo es cubrir la necesidades para representar datos clínicos provenientes de mensajes HL7 versión 2 y 3 en un conjunto de objetos. A partir de éste modelo se ha creado se crea la estructura básica en la que almacenar toda la información clínica en el ámbito de los proyectos INTEGRATE y EURECA.

El "Core Dataset" contiene las ontologías de los vocabularios médicos sobre los que basar el razonamiento semántico, siendo éstas ontologías esquemas conceptuales exhaustivos y completos de dichas terminologías. Además el "Core Dataset" contiene los principales servicios de interoperabilidad semántica que sirven de interfaz entre el modelo común de información y el modelo común de datos. Es en ésta serie de servicios en los que se encuentra tanto el principal objetivo de éste proyecto fin de grado, como uno de los principales desafíos de la interoperabilidad semántica del proyecto.

2.2. Objetivos

El objetivo de éste proyecto fin de grado se encuentra centrado en la generación de un servicio o conjunto de servicios que sirva de enlace entre elementos del modelo común de la información y el modelo común de datos. Con la finalidad de marcar unas breves pautas a seguir para poder conseguir el objetivo de interoperabilidad semántica en el proyecto, se fijan a continuación una serie de objetivos necesarios para lograr el éxito en éste componente fundamental del proyecto:

- **El servicio deberá asignar un concepto dado de una terminología a un elemento de un modelo de datos.** El servicio deberá de ser capaz de dado un concepto pertenciente a SNOMED CT, LOINC y HGNC, vincular dicho concepto a un elemento del modelo de datos, y viceversa. Es decir, dado un elemento del modelo de datos conocer que conceptos pueden encontrarse en dicho elemento.

- **El servicio deberá de proporcionar la integración de un conjunto de terminologías médicas.** El servicio deberá de ser capaz de integrar y homogeneizar datos expresados con diferentes vocabularios médicos. La especificación de estos vocabularios será finalmente SNOMED CT, LOINC y HGNC.

- **Se deberá limitar a un modelo de datos ya existente basado en el HL7 RIM.** Se debe de diseñar el servicio teniendo en cuenta la existencia de un modelo de datos ya implementado y se deberá ceñirse lo máximo posible a ese modelo, tratando de integrar las terminologías establecidas con el menor número de cambios en el modelo de datos.

- **Integración con tecnologías ya empleadas.** Se deberá tener en cuenta las tecnologías ya presentes como MySQL, Sesame, Kettle - Pentaho, Java y SPARQL, y que el servicio se deberá desarrollar intentando usar éstas tecnologías, o que la integración de otras tecnologías no entorpezca el desarrollo del servicio.

- **Alto rendimiento del servicio.** La eficiencia del servicio es algo muy relevante durante su desarrollo. El servicio se trata de una interfaz entre datos expresados en vocabularios médicos muy extensos (SNOMED CT cuenta con más de 300.000 conceptos y 1.300.000 relaciones) y un modelo de datos, y durante el uso del servicio podría conllevar un exceso tiempo de proceso el mero

hecho de acceder a un dato del modelo de datos, si éste implica un razonamiento demasiado extenso sobre una terminología médica.

Los objetivos planteados y sus resultados serán analizados y expuestos a lo largo de éste documento, indicando las decisiones y enfoques finalmente adoptados y la razón de los mismos.

2.3. Organización del proyecto

La presente memoria se estructura de la siguiente manera:

- **Análisis de terminologías médicas y el modelo de datos.** Como objetivo inicial se tendrá que realizar una documentación exhaustiva sobre las terminologías médicas que se van a usar en el proyecto (SNOMED CT, LOINC y HGNC como terminologías médicas principales) y como pueden adaptarse a un modelo de datos basado en el HL7 RIM.

- **Estudio de las tecnologías necesarias.** Una vez analizada las compatibilidades entre terminologías médicas y el modelo de datos, se deberá plantear que tecnologías serán las más adecuadas para el desarrollo del servicio. Teniendo en cuenta las tecnologías ya presentes en el proyecto.

- **Especificación de requisitos y diseño.** Al tratar de ubicarse el servicio que tiene como objetivo éste proyecto dentro de una infraestructura ya desarrollada, se tendrá que tener en cuenta dicha situación a la hora de diseñar el servicio para su total integración con el resto de servicios y componentes. En ésta fase por tanto, se diseñará la arquitectura de componentes funcionales y tecnologías del sistema, así como la arquitectura de los servicios del sistema describiendo su estructura de clases, atributos y relaciones entre ellos.

- **Desarrollo y diseño del prototipo.** Se realizará un primer prototipo de la herramienta, y se definirá tanto el diseño del prototipo como su implementación. Se detallará la implementación de las partes que componen la herramienta desarrollada.

- **Resultados y pruebas.** El principal objetivo de ésta fase final es la revisión del sistema software y que éste cumple realmente con las especificaciones y su cometido. También se analizarán los resultados obtenidos de la aplicación de

la herramienta desarrollada. Durante la elaboración de ésta fase se refinará y planteará una posible línea futura.

- **Conclusiones y líneas futuras.** Finalmente se analizará los logros conseguidos durante el desarrollo del proyecto y se expondrá una serie de aplicaciones futuras o ampliaciones que pueda continuar con el desarrollo del proyecto, o que sirvan como base para el desarrollo de otros nuevos.

Capítulo 3

ESTADO DE LA CUESTIÓN

El trabajo planteado en el presente proyecto se encuadra dentro de dos proyectos de investigación de ámbito europeo, INTEGRATE[5] y EURECA[6], que tienen por objetivo principal la gestión de una gran cantidad de datos muy diversa, procedente de multitud de instituciones en las que se están realizando ensayos clínicos de diferente envergadura. El área de investigación en el que se encuentran estos proyectos es un terreno en el que en los últimos años se ha encontrado un gran desafío a la hora de gestionar tal heterogeneidad de datos, y proporcionar a su vez un acceso homogéneo y con valor semántico[1][2][3][4]. Dotar de ésta capacidad semántica es esencial para para asegurar la sostenibilidad de los datos, ya que de otra manera los usuarios finales se vería forzados a conocer cada modelo y estructura de los datos de cada una de las instituciones de las que quieran recuperar información, proporcionando cada usuario la carencia semántica del sistema al que están accediendo.

Para proporcionar ésta capa de interoperabilidad semántica que posibilite un trabajo colaborativo a diferentes escalas entre investigadores e instituciones clínicas, es necesario el desarrollo de dos componentes bien diferenciados, el "Common Data Model", y el "Core Dataset".

3.1. Common Data Model

Como ya se ha comentado, el objetivo final de los proyectos INTEGRATE y EURECA conlleva la necesidad de crear una estructura capaz de almacenar toda la información que se genera en múltiples instituciones en las que se realizan ensayos clínicos. Ésta estructura debe de ser común ya que debe almacenar toda ésta información de manera homogénea de éstas fuentes diversas. Inicialmente se consideraron diversos modelos como i2b2[7], OMOP[8] y HL7 RIM[9], pero finalmente se

adoptó el Reference Information Model creado por el grupo HL7[10].

3.1.1. HL7 Reference Information Model

Health Level 7 Reference Information Model es un estándar definido por la organización HL7. El objetivo principal de ésta organización es desarrollar un estándar que facilite el intercambio electrónico información clínica, mejorando así la atención en salud, optimizando el flujo de trabajo, reduciendo ambigüedad y mejorando la transferencia de conocimiento entre las partes interesadas, como prestadores de servicios de salud, organismos gubernamentales, etc.

Específicamente, éste modelo usa UML[11] para mostrar un modelo que es capaz de contextualizar cualquier situación relacionada con el entorno de los servicios de salud, desde el diagnóstico efectuado a un paciente, como el material sanitario, el coste de un tratamiento hasta la información del personal sanitario de una institución. El esquema que reúne las clases básicas del modelo se encuentra a continuación en la figura 3.1 mientras que el modelo completo se puede encontrar en el anexo A.2.

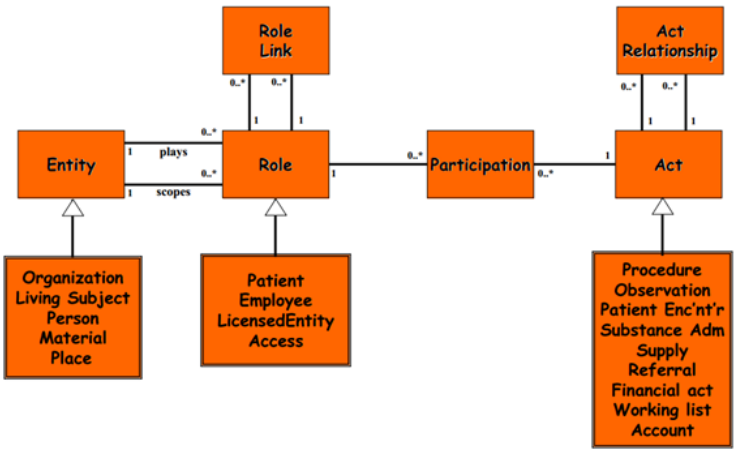

Figura 3.1: Diagrama básico RIM

Como podemos observar, las clases básicas del modelo y en las que nos centraremos principalmente durante éste proyecto. Las tres clases básicas serían por tanto:

- **Act**: Act indica la acción que haya tenido lugar, esté teniendo lugar, vaya a tener lugar, haya sido solicitada o se tenga planeado que ocurra. Cada instancia

de acto representa un hecho en un punto concreto del tiempo, y todo acto debe tener algún participante.

- **Role**: Role establece el papel o cometido de una entidad en un acto.
- **Entity**: Entity representa cualquier ente, desde sujetos vivos hasta sustancias químicas o biológicas, que pueda participar en determinado acto.

A su vez, como se puede apreciar en la figura 3.1 y con mayor detalle en el anexo A.2, la clase Act puede subdividirse en diversas categorías o tipos. Explicar cada una de estas categorías es fundamental, ya que las diferencias existentes entre las mismas son muy relevantes para la consistencia de los datos a integrar. Las subclases de acto, poseen también atributos únicos que aportan atributos adicionales para poder expresar más precisamente un determinado acto. Las subclases más relevantes para el presente proyecto son las que siguen:

- **Observation**: Especifica que la acción realizada se trata de un acto de reconocimiento o indicación de determinada información sobre un sujeto y otras entidades relacionadas. Puede ser mediciones, métodos de investigación o meras declaraciones asertivas.
- **Procedure**: Indica que la acción realizada en el sujeto se trata de algún tipo de intervención o manipulación de parte o partes de su cuerpo.
- **Substance Administration**: A pesar que se trata a su vez de una subclase de "Procedure", éste tipo de acción se refiere a la acción de introducir o aplicar una determinada sustancia o compuesto a un sujeto.

Cada una de las listadas subclases de acto, poseen a su vez otras subclases, como el caso de "Procedure" y "Substance Administration", pero gran parte de la carga semántica que soporta éste modelo viene dada por los atributos de dichas clases. Alguno de los atributos más representativos de la clase acto serían: "classCode", que especifica la naturaleza del acto; "moodCode", que diferenia si un acto se concibe como una delcaración de hecho o de alguna manera como un comando, posibilidad u objetivo; "statusCode" que define el estado de un acto, si se encuentra activo, completado, cancelado o suspendido; y finalmente "actionNegationInd" que indica si el acto es una negación del acto en sí.

3.2. Core Dataset

Para poder conseguir una verdadera interoperabilidad semántica y poder unir sin problemas infraestructuras clínicas y de investigación ya existentes, como sistemas

de ensayos clínicos, eCRFs y EHRs, que son expresados mediante diversos métodos y terminologías, se requiere de un núcleo de datos básico de los vocabularios que usan para poder realizar el razonamiento necesario para conseguir la interoperabilidad entre estos sistemas. Los vocabularios, como ya se ha mencionado, son SNOMED CT[12], LOINC[13] y HGNC[14], y las organizaciones y entidades que los mantienen y gestionan proporcionan los medios necesarios para poder generar las ontologías que nos permiten el desarrollo de nuestras herramientas. El "Core Dataset", aparte de incluir estos datos básicos procedentes de las diferentes "lenguas" usadas entre los múltiples sistemas, contendrá los mecanismos necesarios para comportarse como el principal motor desambiguador o torre de babel de todo el proyecto.

3.2.1. Teminologías Médicas

Para poder expresar información clínica, se requiere del uso de una serie de términos muy específicos para poder describir con precisión el cuerpo humano y sus asociadas condiciones, componentes y procesos. A lo de largo de la expansión del conocimiento sobre el área médica, la terminología usada para describir tal conocimiento se ha ido expandiendo siguiendo unas normas y reglas lingüísticas. En las últimas décadas se han ido desarrollando una serie de vocabularios específicos que intentan englobar y contener toda la expresividad requerida para determinar hasta el más específico proceso del cuerpo humano. El avance de las tecnologías de la información han condicionado a estos vocabularios adaptarse y facilitar su gestión mediante estas tecnologías, y es así como finalmente han ido evolucionando terminologías como SNOMED CT, en las que los términos médicos que representan están estructurados siguiendo jerarquías y el uso de sistemas de codificación.

3.2.1.1. SNOMED CT

Se trata de una colección de términos médicos sistemáticamente organizados que permiten ser procesados por sistemas de la información. Su objetivo principal es proporcionar un registro de datos clínicos efectivo, mejorando así el cuidado del paciente. Ha sido desarrollado desde una nomenclatura centrada en patología (SNOP) hasta terminología médica lógica. En enero del 2002 SNOMED CT fue creada para unir y expandir la terminología SNOMED RT del Colegio Americano de Patologistas (CAP) con "Clinical Terms" del Servicio Nacional de Salud de UK (NHS). Actualmente, la "National Library of Medicine" (NLM) mantiene un acuerdo para perpetuar la licencia de SNOMED CT y futuras actualizaciones, y permitiéndoles una perpetua licencia para distribuir SNOMED para su "Unified Medical Language System" (UMLS). Actualmente es actualizado y distribuido por "The International Health Terminology Standards Development" (IHTSDO), ya que adquirió ésta

terminología en Abril del 2007.

SNOMED CT incluye más de 310.000 conceptos identificados por códigos únicos, y estos conceptos están relacionados entre sí por más de 1.360.000 relaciones. Su estructura se trata de una organización jerárquica de clases, lo que quiere decir que determinado término más o menos específico, es incluido dentro de la súper clase de otro término que representa un significado más generalista. A parte de las relaciones que estructuran la terminología como una jerarquía de clases (relaciones is-a), existen otras relaciones que dotan a SNOMED CT para que pueda ofrecer un mayor significado, como relaciones descriptivas para proporcionar información acerca de la ubicación, causa, severidad o temporalidad de determinado término.

Gracias a la sintaxis compositiva que ofrece SNOMED CT, ante la ausencia de determinado concepto para alguna acción, ésta puede ser descrita mediante un conjunto de conceptos ya presentes en el vocabulario. Éste tipo de conceptos, compuestos por la composición de algunos otros, se le denomina pos coordinación, y es una de las capacidades que otorgan a SNOMED CT de una mayor expresividad comparado con otras terminologías. Cuando algunas de estas composiciones se incluyen en algunas de las revisiones de SNOMED CT, atribuyéndole un concepto y código dentro de la terminología, se le denomina pre coordinación.

Una de las grandes deficiencias de ésta terminología es la falta de compromiso ontológico, ya que muchas veces no queda claro que tipo de entidad es instancia de un concepto dado, como por ejemplo, intentar determinar si un tumor hace referencia a un proceso o a un trozo de tejido. Ésta ambigüedad que nos encontramos al hacer uso de ésta terminología dificulta enormemente la labor de mantener una consistencia y desambigüedad con los datos que vayan a ser integrados en nuestra infraestructura.

3.2.1.2. LOINC

"Logical Observation Identifiers Names and Codes"[13] es una de las terminologías de uso más extenso para representar identificar observaciones de laboratorios médicos. Fue creado y desarrollado por el Instituto Regenstrief en 1994. Originalmente fue creado como respuesta a la demanda para una base de datos clínica que no supusiera coste alguno.

Cerca de 58.000 observaciones están incluidas en LOINC, formadas 6 componentes o campos con los que pueden determinar una única medición u observación especificando: el componente que es medido, evaluado u observado; el tipo de característica medido; el intervalo de tiempo en que fue realizado; el contexto o espécimen en el que fue realizado; la escala de medida; procedimiento usado para realizar la medición u observación.

Su finalidad por tanto es la de facilitar el intercambio y la integración de resultados clínicos procedentes de test de laboratorios, observaciones clínicas y de investigaciones. Una de las ventajas de usar LOINC es que gracias a que se trata de una terminología tan específica, consigue representar de manera inequívoca determinada acción realizada sobre un paciente, pero obviamente esto conlleva que LOINC necesita de una armonización con SNOMED CT, ya que no es suficientemente amplio como para expresar una generalidad tal y como se encuentra desarrollado SNOMED CT.

3.2.1.3. HGNC

El HUGO Gene Nomenclature Committee[14] establece un único nombre para cada uno de los genes humanos conocidos, asignándole a cada gen un nombre largo y una abreviación o símbolo. Hasta el día de hoy han sido representados más de 35.000, siendo 19.000 de ellos la codificación de proteínas.

El propósito de ésta terminología es proporcionar una representación específica de cada gen conocido. Para conseguir esto, HGNC intenta ponerse en contacto con los investigadores que hayan publicado algún gen durante su investigación proponiéndoles una nomenclatura para dicho gen, para que pueda ser representado e incluido en HGNC.

3.2.2. Normalización SNOMED CT

La forma normal de SNOMED CT[15] es una vista que puede ser generada por cualquier expresión válida aplicando un conjunto de transformaciones lógicas[15]. Un concepto puede ser primitivo o encontrarse totalmente definido, cuando sus roles o parientes no expresan completamente su significado. Los conceptos primitivos no tienen las relaciones que les permiten ser diferenciados de sus antecesores o de sus predecesores ya quue los conceptos completamente definidos puede ser diferenciados por el significado de sus relaciones. Por ejemplo, “Neoplasm of breast” tiene dos relaciones, “Associated morphology” con un valor de “Neoplasm”, y un “Finding site” con un valor de “Breast”, mientras que los antecesores y predecesores de cáncer de pecho tienen diferentes relaciones o diferentes valores de las relaciones, por lo que no sería un concepto completamente definido.

Una de las de las principales características de SNOMED CT es que todo concepto que resulta de la forma normal de otro concepto, será primitivo. Durante el

desarrollo de la forma normal, se pueden diferencian dos formas de expresar la forma normal de un determinado concepto:

- **Forma normal larga** Establece que todos los atributos que pueden ser inferenciados desde los conceptos referenciados por la expresión.
- **Forma normal corta** Permite una manera más eficiente de recuperar la información relacionada con la forma normal, ya que no incluye los atributos de la primitiva más próxima al concepto que se está normalizando, si no tan solo los atributos que definen el concepto que se está normalizando.

Debido a que la forma normal corta proporciona información suficiente y la requerida para el procesado de determinado concepto, ya que mantiene toda la información que realmente se necesita para poder integrar cualquier concept de SNOMED CT, se usará éste tipo de formato de normalización.

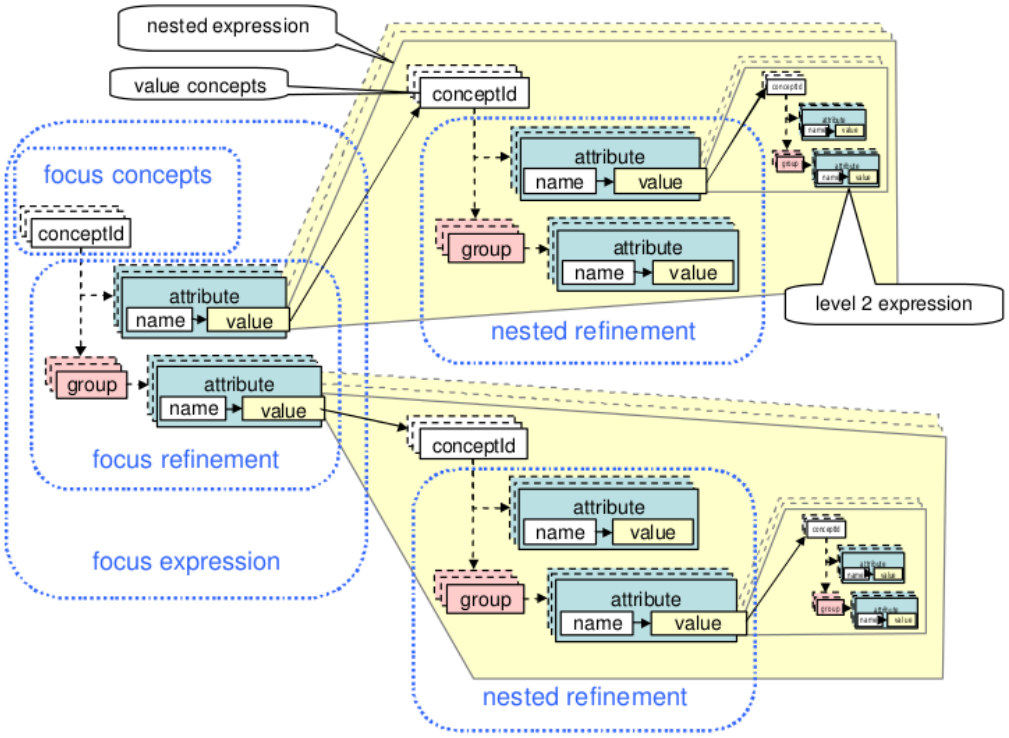

Figura 3.2: Estructura de la forma normal[26]

Una expresión en forma normal corta de SNOMED CT puede ser dividida en dos partes:

- **Focus concept** Es el concepto antecesor primitivo más próximo al concept que se está normalizando.
- **Refinamiento** A través del refinamiento se extraen las relaciones que definen un concepto. Éstas relaciones puede ser agrupadas en cero, uno o más "role groups". Cada "role group" es creado para especificar una expresión permitiéndola agrupar las relaciones de una manera más estructurada. Las relaciones están formadas por pares de conceptos, siendo uno un concepto atributo, con un significado sintáctico, y otro el valor del atributo, que es el que proporciona un mayor significado al concepto de que es relación.

Para poder comprender mejor el proceso de composición de la forma normal, se puede visualizar dicha composición en la figura 3.2.

3.3. Proyecto HL7 TermInfo

El TermInfo[26] es un proyecto desarrollado por el grupo HL7 y que intenta ofrecer una serie de directrices para el uso de terminologías en modelos de información, siendo el modelo HL7 RIM, y la terminología SNOMED CT los principales protagonistas de éste proyecto.

El proyecto TermInfo nació como un proyecto lanzado por la NASA en Julio de 2004, con la finalidad de permitir la codificación y estructuración de los registros clínicos de los astronautas. Debido a que la información clínica estaba principalmente representada usando la terminología SNOMED CT, y el modelo de información elegido fue el HL7 RIM, el proyecto TermInfo se lanzó inicialmente como el estudio del uso de SNOMED CT sobre HL7 RIM. Teniendo como objetivo el proporcionar significado a la información estructurada de una terminología médica.

La importancia del proyecto TermInfo radica en que HL7 RIM provee de una estructura que puede ser poblada con instancias de información que es representada usando un vocabulario codificado. SNOMED CT es una terminología de referencia que proporciona un vocabulario codificado para poblar un modelo de información con conceptos procesables. Juntando ambas necesidades nos encontramos con el marco adecuado para poder cumplir con el objetivo final de la interoperabilidad semántica, que es compartido tanto por HL7 como por SNOMED. La base de ésta interoperabilidad es que el significado depende de cómo el vocabulario y la estructura encajan juntas.

El principal problema que se encuentra es que el significado depende del contexto de las palabras dentro de una oración. Un ejemplo de esto sería:

- Los resultados sugieren la exclusión de tirotoxicosis.
- Los resultados sugieren que tirotoxicosis puede ser excluida.
- Los resultados sugieren que tirotoxicosis debería de ser excluida.
- Los resultados sugieren que tirotoxicosis necesitaría ser excluida.

Como se puede observar, el significado de cada oración es diferente, e implicaría una información diferente en cada caso, tomando unas medidas u otras dependiendo de la oración.

Para poder aclarar esto, es donde surge e interviene TermInfo, ya que intenta proporcionar una serie de ejemplos de recomendaciones a seguir. Las recomendaciones que proporciona, a veces son de carácter general, indicando por ejemplo que la interoperabilidad semántica requiere de un modelo compartido de significado, o recomendaciones más precisas como que se evite el uso de algunos atributos del HL7 RIM que restan significado a expresiones de SNOMED CT. En otras ocasiones, en vez de recomendaciones, expresan advertencias de posibles solapamientos semánticos entre tablas del HL7 RIM y partes del vocabulario SNOMED CT.

Un ejemplo de éste tipo de solapamiento semántico sería el uso del atributo moodCode de la clase Act, que puede tomar el valor de “INT” (intención o propósito) junto con el contexto de procedimiento de SNOMED CT “requested” (solicitado), que usando ambos cambios podría llegar a entender erróneamente como “intención de solicitar”, cuando realmente se trata simplemente de que el acto ha sido solicitado, y ahí viene implícito la intención del acto.

Capítulo 4

TECNOLOGÍAS USADAS

4.1. Java

La mayor parte de la codificación que se ha realizado en éste proyecto se ha efectuado en el lenguaje de programación Java. Una de las razones de ello, es que era el lenguaje que ya se estaba usando en el ámbito del proyecto para ofrecer los servicios webs de la capa semántica del proyecto.

Es un lenguaje de programación publicado en 1995 y desarrollado por James Gosling de Sun Microsystems, actualmente adquirida por Oracle. Es un lenguaje de propósito general, concurrente, basado en clases y basado en clases. Su sintaxis deriva mucho de los lenguajes C y C++, pero con menos facilidades a bajo nivel, lo que dependiendo del desarrollador, puede facilitar su manejo, o restringir parcialmente la flexibilidad o potencial del lenguaje. Su principal punto fuerte es que sus aplicaciones son compiladas a un bytecode que puede ser ejecutado en cualquier máquina virtual Java (JVM), y debido a que ésta máquina virtual puede correr sobre múltiples arquitecturas, una aplicación java en bytecode puede ejecutarse casi indiferentemente de la arquitectura del computador. Éste gran beneficio requiere un consumo extra de recursos para la interpretación del programa, conllevando que la ejecución sea más lenta que un ejecutable nativo.

Otra de las características fundamentales de Java es relativa a su método de programación y diseño orientado a objetos. La idea esencial de tal diseño de programación consiste en diseñar el software de forma que los distintos tipos de datos que se usen estén unidos a sus operaciones, combinando los datos y el código en entidades llamadas objetos. El diseño orientado a objetos permite contener en un solo objeto el paquete que está formado tanto por el comportamiento o código, y por el estado o datos. Separando de ésta manera objetos independientes proporciona

una base más estable para el diseño de un sistema software, ya que al poder tratar un objeto como una unidad independiente, éste objeto puede gestionarse de manera más fácil. Además se permite incluso la reutilización del software contenido en un objeto para otros proyectos o programas, siempre que estos objetos sean diseñados o se cree específicamente un objeto más genérico. De ésta manera, se podría considerar cada objeto como una pieza de software reutilizable, proporcionando una modularidad a los sistemas software desarrollados con java que propician una reducción en el tiempo de desarrollo y en la calidad del software generado.

La sintaxis de Java deriva principalmente de C++, pero construida exclusivamente para la programación orientada a objetos, ya que en Java, como ya se ha comentado, todo es un objeto salvo algunas excepciones. Para apreciar mejor el diseño orientado a objetos y su sintaxis podemos disponer del siguiente código del presente proyecto:

```
1  static class MapValue {
2      String table;
3      String attribute;
4      Boolean checked;
5
6      public MapValue(String t, String a, Boolean c){
7        table = t;
8        attribute = a;
9        checked = c;
10     }
11
12   }
13
14 static Map<Integer, MapValue> mp = new HashMap<Integer, MapValue>()
       ;
15
16 public static void main(String[] args){
17
18      CoreDatasetService st = new CoreDatasetService();
19
20      mp.put(404684003, new MapValue(``Observation'',``code'', true)
21
22      st.CD2CDM(mp.get(404684003));
23
24
25 }
```

Figura 4.1: Ejemplo sintaxis y orientado a objetos Java

Como podemos observar en 4.1 el método principal al que lanzador de Java pa-

será el control del programa es el iniciado por el nombre "main". Como se puede observar, se está haciendo uso de diferentes objetos, que en el caso de la clase "MapValue", se puede observar cómo se declara ese objeto, así como parte del código de inicialización de dicho objeto, que luego será usado durante la ejecución del programa. Además, como ya se comentó anteriormente, el diseño basado en objetos de Java permite el uso de objetos ya elaborados, como ese el caso en 4.1 cuando se da uso de la clase CoreDatasetService y da uso a uno de sus métodos, estando dicha clase definida a parte del programa que se ha mostrado.

Finalmente, otra de las grandes ventajas de Java es la inercia y fuerza que posee actualmente en el mundo software su comunidad de desarrolladores que proporcionan gran cantidad de recursos, disponibles tanto en librerías como en documentación. Lo cual facilita enormemente el desarrollo usando el lenguaje Java.

4.2. Web Services

Una de las necesidades principales del proyecto es disponer de una serie de servicios que sirvieran de capa semántica para conseguir la interoperabilidad semántica. Los servicios a desarrollar deberían de ser accesibles por cualquiera del resto de miembros de grupo de trabajo del proyecto, para que puedan hacer uso de la capa semántica en las partes del proyecto que estén desarrollando, como puedan ser los proveedores de datos o la plataforma de selección de criterios de elegibilidad.

Para proporcionar estos servicios se está haciendo uso del método de comunicación Web Service a traves de la World Wide Web, que proveen de función software en una dirección de red determinada. Como se define el World Wide Web Consortium[17], un servicio web es un software diseñado para sostener la interoperabilidad de interacción entre dispositivos sobre la red. Para poder interaccionar con un servicio desplegado en una máquina, éste dispone de una interfaz que en un formato procesable, en WSDL[18], contiene toda la información necesaria para poder comunicar con dicho servicio. La comunicación que se efectúa entre la máquina cliente y la máquina servidora que dispone del servicio, se da uso de mensajes SOAP[19], que son normalmente transmitidos mediante HTTP con una serialización XML en conjunto con otras normas relacionadas con la web.

Una de las ventajas de ésta tecnología consiste en a través de la descripción de las operaciones que se ofrecen en un servicio proporcionada por el "Web Services Description Language"(WSDL) se puede generar de manera automática tanto el código del cliente, como el esqueleto del código del servidor. De ésta manera se

ofrecen dos tipos de estrategias de diseño a la hora de generar un "Web Service":

- **Estrategia bottom-up:** El diseño bottom-up consiste en que el desarrollador implemente las clases del servicio primero, y una vez generadas las clases o código del servidor, se hace uso de una herramienta que genere el WSDL de dicha implementación de manera automática. Así se facilita el desarrollo pero puede ser mucho más difícil de mantener si las clases originales están sujetas a múltiples cambios.

- **Estrategia top-down:** El diseño top-down consiste en que lo primero que se acuerda es el diseño del WSDL, y a través de éste WSDL con una herramienta automática se crea el "skeleton" de la clase del Web Service, y a partir de ésta base se implementa el servidor. El diseño top-down hace más difícil el desarrollo, pero produce unos diseños mas limpios que generalmente son más resistentes a los cambios.

En el presente proyecto, se ha estado haciendo uso de la estrategia bottom-up, ya que al tratarse de un proyecto de investigación no existe gran necesidad de acordar un WSDL, sobre el que otras partes fundamenten su desarrollo, por lo que si se decide cambiar el WSDL de los Servicios Web que se están desarrollando no supone apenas ningún inconveniente. Pro tanto, mientras que un diseño top-down nos podría limitar nuestras posibilidades en el desarrollo de nuestro proyecto, el diseño bottom-up nos permite gran flexibilidad y facilidades sin apenas trabas.

4.2.1. Apache Tomcat

Como se ha comentado, la labor de éste proyecto es el elaborar un servicio web que sustente parte de la capa semántica necesaria para lograr la interoperabilidad entre un modelo de datos y términos médicos. Para poder desplegar los servicios necesarios para conseguir dicho fin se requiere de una herramienta que los contenga, en el caso de nuestro proyecto se está haciendo uso del Apache Tomcat[20].

Tomcat es un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat provee de un servidor web HTTP en cuyo entorno puede ejecutar código Java, ya que implementa las especificaciones de Java Servlet y JavaServer Pages (JSP). Tomcat está formado por una serie de componentes con diferentes finalidades, algunos formaban parte de Tomcat desde un inicio como Catalina, Coyote y Jasper, pero otros fueron agregados en versiones posteriores como Cluster. Aunque se podría considerar Catalina como el componente principal ya que implementa las especificaciones de Sun Microsystems para serverlet

y JavaServer Pages, mientras que Coyote se encarga de las proporcionar los conectores necesarios, y Jasper del motor JSP compilándolos y convirtiéndolos en servlets.

Una de las ventajas de Apache Tomcat radica en que está implementado en Java, por lo que puede ejecutarse en cualquier plataforma en la que se pueda instalar una máquina virtual Java. Además hoy día Tomcat es usado como un servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad, gracias a uno de los componentes añadidos durante las múltiples versiones del Tomcat. Esto se consigue, entre otros medios, gracias a que Tomcat replica el servidor web en uno temporal en un puerto diferente al que reenvía todas las peticiones entrantes, mientras que realiza labores de actualización del servidor en el puerto principal, proporcionando así una alta disponibilidad del sistema.

Apache Tomcat en sí, no proporciona la capacidad de agregación y despliegue de servicios web, pero sin embargo si dispone de un motor adecuado para dicho fin mediante Apache Axis2.

4.2.2. Apache Axis2

Axis2 es un motor para servicios web que es un completo rediseño del ampliamente usado Apache Axis SOAP. No solo provee de capacidades de añadir interfaces de servicios web a aplicaciones web, si no también puede funcionar como una aplicación server independiente. Axis2 soporta SOAP 1.1 y SOAP 1.2, pero también incluye soporte para los servicios web REST.

Una de las principales características por las que hacer uso de Axis2 es que permite “Hot Deployment”. El despliegue en caliente permite la capacidad de desplegar servicios mientras que el sistema se encuentra ejecutando, facilitando así el desarrollo y mantenimiento de los servicios implementados, ya que evita el apagado y encendido de toda la plataforma cada vez que se quiera agregar o actualizar un servicio web. Otras de las ventajas más relevantes al usar Axis2, siendo de relevante importancia, es la velocidad que proporciona. Axis 2 usa su propio modelo de objetos y análisis “StAx” (Streaming API for XML), que proporciona una gran velocidad comparado con versiones anteriores de Apache Axis. Ésta característica posibilita un flujo mayor de datos y a mayor velocidad entre el cliente servidor, lo que cual es fundamental teniendo en cuenta los grandes volúmenes de datos que se generan durante el razonamiento semántico de algunos términos médicos.

4.3. Resource Description Framework

El Marco de Descripción de Recursos (RDF) es un framework para metadatos de la World Wide Web (WWW), desarrollado por el World Wide Web Consortium (W3C)[17]. En sí, se trata de un lenguaje objetivo general para la representación de información en la web mediante una descripción conceptual. Actualmente está siendo usado como método general para la descripción conceptual o modelado de la información e implementado con recursos web, usando una variedad de sintaxis y anotaciones.

El modelo de datos RDF es similar a modelos consistentes en entidad-relación o diagramas de clases, ya que está basado en la idea de hacer declaraciones sobre recursos web con expresiones con forma de sujeto-predicado-objeto conocidas como tripletas en la terminología RDF. El sujeto indica el recurso, mientras que el predicado denota rasgos o aspectos del recurso y expresa una relación entre el sujeto y el objeto.

Éste mecanismo es uno de los principales componentes en la actividad de la web semántica del W3C, presentando un estado evolutivo de la World Wide Web en el que software automatizado puede almacenar, intercambiar y usar información distribuida por la web, permitiendo a los usuarios el manejo de información con mayor eficiencia y seguridad. Hay que tener en cuenta que para ciertas áreas de conocimiento un modelo de datos basado en RDF es más recomendable que modelos relacionales u otros modelos ontológicos. Aunque a menudo los datos RDF pueden persistir en bases de datos relacionales con representaciones nativas.

Para poder acceder o consultar a la información por éste modelo es recomendable el uso de algunos lenguajes de consulta como pueden ser SPARQL, SerQL o Versa. Estos lenguajes en muchos casos no sólo sirven para poder acceder a la información almacenada en formato RDF, si no en también manipular y gestionar los datos almacenados de ésta manera, sirviendo por tanto de interfaz entre el usuario o aplicación y los datos.

4.3.1. OWL

El Ontology Web Language es un lenguaje desarrollado por el grupo de trabajo "Web Ontology" del W3C[17]. Es un lenguaje para la representación de ontologías, es decir, para la especificación de conceptos, relaciones entre conceptos, individuales y axiomas. Permite a los usuarios definir clases, asociarlas propiedades, y definir relaciones entre ellas. Los conceptos en "OWL" pueden ser definidos basados en

la unión. intersección y complemento de otros conceptos. De manera adicional, el lenguaje permite la especificación de relaciones de equivalencia y disyunción entre conceptos. En términos de poder expresivo, "OWL" tiene una mayor capacidad expresiva que "RDF". La base formal de OWL es la familia de formalismos de representación del conocimiento conocido como lógicas descriptivas.

4.4. SPARQL

"SPARQL Protocol and RDF Query Language" es un lenguaje de consulta RDF para recuperar y manipular datos almacenados en formato RDF. Se trata de un estándar creado por el "RDF Data Access Working Group" del W3C[17], siendo considerado como una de las tecnologías claves para la web semántica.

Como ocurre con SQL, hay que diferenciar entre el lenguaje de consulta y el motor de almacenamiento y recuperación de datos. Por ésta razón existen múltiples implementaciones de SPARQL para diferentes plataformas tecnológicas y entornos de desarrollo.

El formato de las consultas en SPARQL se puede dividir en diferentes variaciones dependiendo del propósito:

- **SELECT** Usado para extraer valores brutos en forma de tabla de un endpoint SPARQL.
- **CONSTRUCT** Usado para extraer información de un endpoint SPARQL y transformar los resultados en formato RDF válido.
- **ASK** Usado para proporcionar un simple Verdadero/Falso para una consulta dada.
- **DESCRIBE** Usado para extraer un grafo RDF de un endpoint SPARQL, cuyo contenido se deja a criterio de valoración para decidir sobre si la información recuperada es útil.

Todos estos posibles formatos de consultas en SPARQL incluyen un bloque **WHERE** con el que restringir los posibles valores a devolver en la consulta.

4.5. Sesame

Sesame es un framework de código abierto basado en Java para el almacenaje, procesamiento y consulta de datos RDF. Fue desarrollado como prototipo de aplicación para el proyecto Europeo "On-To-Knowledge" y es actualmente mantenido por

la compañía holandesa Aduna. A lo largo de los años Sesame se ha convertido en una de las plataformas de facto para la gestión de metadatos y datos RDF, característica que se refleja en el número de otras tecnologías semánticas que hacen uso de ella.

La arquitectura del sistema sigue un diseño por capas. Cada capa implementa un conjunto de servicios que pueden ser accedidos por las capas superiores. La arquitectura fue diseñada para permitir que parte de la funcionalidad sea implementada por componentes externos como razonadores. Cada uno de los niveles sería:

- **Modelo RDF** Es la capa inferior y sobre la que todo el framework Sesame está construido. Aquí se implementa la representación del modelo RDF y provee de métodos para el manejo de diferentes entidades RDF.
- **Capa RDF I/O** Provee de los métodos a bajo nivel para la lectura y escritura de documentos RDF.
- **Capa SAIL** Es la capa de "Storage and Inference Layer", y es una API que proporciona abstracción respecto de la tecnología o sistema de almacenamiento. Implementa una serie de métodos para el acceso al almacenamiento RDF y los motores de inferenciación.
- **Cliente HTTP** A partir de la versión v2.x se implementa un cliente HTTP que proporciona a los usuarios con métodos de acceso a los repositorios Sesame que se encuentren en ejecución tras los servidores HTTP.
- **API del repositorio** Es una API de alto nivel que implementa una serie de mecanismos para el manejo de datos RDF, permitiendo a los desarrolladores tener un acceso simple a los métodos de lectura, consulta y gestión de datos RDF.
- **HTTP Server** Sesame implementa un servidor HTTP que permite a los usuarios el establecer una serie de repositorios en un servidor HTTP para poder acceder por HTTP. Ésta funcionalidad está implementada usando Java Servlet y JSP.
- **Aplicaciones de Usuario** Aplicaciones de dominio específico que utilizan el framework Sesame para establecerse en la capa superior de la API del repositorio.

Sesame puede ser extendido y configurado para sostener una variedad de tecnologías y estándares relacionados con diversos aspectos. Sesame permite la serialización RDF, compatible con formatos como XML, TRIG, TRIX, Turtle, N-Triples y N3; Tecnologías de almacenamiento en memoria o en disco; Inferencia, permitiendo

diferentes formas de razonamiento; Diferentes lenguajes de consulta como SPARQL y SerQL; Almacenamiento de contexto e información de origen; Soporte para un protocolo REST-full; Soporta acceso transaccional para datos RDF.

4.6. MySQL

Para la labor del proyecto y desarrollo del modelo común de datos se requiere de un sistema gestor de base de datos. La tecnología que se está haciendo uso es de MySQL, el cual es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Desarrollado por Sun Microsystems en C y C++ como un sistema multiplataforma.

MySQL es un sistema de administración o gestor de bases de datos que maneja la información estructurada en tablas de éstas bases de datos. A raíz de la necesidad intrínseca de la computación del manejo de grandes volúmenes de datos surge la necesidad de gestionar estos datos de manera rápida y flexible. Para ello, estos datos se archivan en tablas separadas en vez de en una única tabla. Además estas tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas, creando así el sistema relacional de bases de datos.

Las características a destacar en MySQL serían:

- Dispone de un amplio subconjunto del lenguaje SQL así como algunas de sus extensiones.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones, etc.
- Transacciones y clave foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto

Sin embargo, la principal característica por la que adoptar ésta tecnología para el ámbito del proyecto sería en que es un sistema software de fuente abierta, lo cual conlleva entre otras cosas que se ofrezca bajo una licencia GNU GPL. Esto

nos permite usar ésta tecnología de manera libre siempre y cuando no derive en un producto privativo, en cuyo caso se debería de comprar la licencia específica. Al no ser éste el caso, podemos usar MySQL en el ámbito del proyecto sin encontrarnos con problemas de escalabilidad y coste que supondría el uso de otra herramienta.

4.7. Mensajes HL7 v2 y v3

Los datos clínicos facilitados para la labor del proyecto han sido proporcionados principalmente usando los estándares del grupo HL7 v2 y v3. Estos estándares proveen las especificaciones que establecen el lenguaje, la estructura y los tipos de datos requeridos para construir los mensajes que sirvan como medio de las transacciones entre las partes involucradas.

Debido a que HL7 International es una organización de reconocido prestigio y que ha creado múltiples estándares en el entorno clínico y de la salud, ha originado que diversas compañías usen estos estándares. Esto facilita en cierta medida la labor de investigación y desarrollo de proyectos en éste área, al incentivar la creación de herramientas que usan estos estándares.

Uno de los problemas existentes al usar ésta tecnología es la versión del estándar en el que se encuentre proporcionada la información. Cada una de estas versiones tiene unas especificaciones y un formato casi completamente diferente una de la otra. Que se detallará en las siguientes secciones.

4.7.1. HL7 v2

Los mensajes HL7 v2 es un estándar que se basa en un flujo legible de segmentos y delimitadores. Ésta sintaxis no está basada en un común XML y tiene su propia sintaxis.

La sintaxis de un mensaje HL7 v2 consiste en uno o más segmentos que son separados por el carácter retorno de carro y consecuentemente cada segmento se dispone en una diferente línea de texto. Cada segmento se divide en uno o más campos separados por el carácter de barra vertical, y cada uno de estos campos se subdivide a su vez en componentes separados por el carácter de acento circunflejo y estos a su vez en subcomponentes separados por ampersand. El carácter de tilde ( ) es por defecto el separador de repetición. Si un mensaje contiene un carácter delimitador especial, existen establecidos una serie de secuencias de caracteres de escape para su representación. El nombre del segmento es especificado por el primer campo del segmento, el cual es siempre un código de 3 caracteres (Message Header

MSH, Patient information PID, Next of Kin NK1, Patient Visit PV1, etc) que identifica el tipo de mensaje. Un ejemplo de dicho mensaje se puede encontrar a continuación en la figura 4.2.

```
MSH|^~\&|MegaReg|XYZHospC|SuperOE|XYZImgCtr|20060529090131
-0500||ADT^A01^ADT_A01|01052901|P|2.5 EVN||200605290901|||
|200605290900 PID|||56782445^^^UAReg^PI||KLEINSAMPLE^BARRY
^Q^JR||19620910|M||2028-9^^HL70005^RA99113^^XYZ|260 GOODWI
N CREST DRIVE^^BIRMINGHAM^AL^35 209^^M^NICKELL'S PICKLES^1
0000 W 100TH AVE^BIRMINGHAM^AL^35200^^O |||||||0105I30001^
^^99DEF^AN PV1||I|W^389^1^UABH^^^^3||||12345^MORGAN^REX^J^
^^MD^0010^UAMC^L||678 90^GRAINGER^LUCY^X^^^MD^0010^UAMC^L|
MED|||||A0||13579^POTTER^SHER MAN^T^^^MD^0010^UAMC^L||||||
||||||||||||||||||||200605290900 OBX|1|NM|^Body Height||1
.80|m^Meter^ISO+|||||F OBX|2|NM|^Body Weight||79|kg^Kilogr
am^ISO+|||||F AL1|1||^ASPIRIN DG1|1||786.50^CHEST PAIN, UN
SPECIFIED^I9|||A
```

Figura 4.2: Ejemplo de la sintáxis de un mensaje HL7 v2

4.7.2. HL7 v3

El estándar HL7 v3 es al contrario que la versión anterior, su sintaxis sigue una codificación basada en XML. La estructura de dicho mensaje vendría orientada por la "HL7 Clinical Document Architecture" (CDA), cuyo propósito es especificar el formato, la estructura y la semántica de documentos clínicos. El esquema XML para cualquier mensaje debe tener una estructura mínima para poder satisfacer el estándar CDA para el intercambio de datos clínicos:

- **Cabecera**
  - Identificación del esquema CDA
  - Elementos del documento clínico: typeID, id, cod, effectiveTime, confidentialityCode
  - recordTarget
  - author
  - custodian
  - component

- **Cuerpo**
  - Para un cuerpo estructurado: component - CuerpoEstructurado - component - section
  - Para un cuerpo no estructurado: component - CuerpoNoEstructurado - text

Para poder asentar un poco la estructura de éste estándar, se dispone un ejemplo en el que se puede comprobar la estructura de un mensaje HL7 v3 en la figura 4.3.

```
<section>
  <code code="19790-5" codeSystem="2.16.840.1.113883.6.1"
     codeSystemName="LOINC" />
  <title>Medication Administration</title>
  <text>
  <list>
    <item>Lorazepam</item>
  </list>
  </text>
  <entry>
    <substanceAdministration classCode="SBADM" moodCode="EVN">
      <text>Lorazepam</text>
      <consumable>
        <manufacturedProduct>
          <manufacturedLabeledDrug>
            <code code="387106007" codeSystem="
               2.16.840.1.113883.6.96"
               codeSystemName="SNOMED CT" displayName="Lorazepam (
               substance)" />
          </manufacturedLabeledDrug>
        </manufacturedProduct>
      </consumable>
    </substanceAdministration>
  </entry>
</section>
```

Figura 4.3: Ejemplo de la sintáxis de un mensaje HL7 v3

A parte de los elementos mínimos que son requeridos el estándar CDA y que tiene que cumplir la estructura de un mensaje HL7 v3, también se debe de disponer de un esquema XML con el que poder validar cualquiera de estos mensajes. Éste esquema, puede ser adquirido por su compra a la organización HL7 o puede ser extraído parcialmente a través del análisis de mensajes ya validados. Para el presente proyecto se ha usado un esquema generado de ésta manera, debido al coste que supondría la adquisición de dicho esquema directamente de la organización.

4.8. Herramientas Extract, Transform and Load

El objetivo de éstas tecnologías ETL es el de extraer, transformar y cargar datos de una fuente a sistema de almacenamiento nuevo. Durante éste proceso los datos pueden o deben ser transformados para poder adaptarse a la estructura del sistema de almacenamiento objetivo. A éste mecanismo se le puede denominar proceso de integración de datos.

La finalidad de éste proyecto es la de incluir el servicio TermBinding dentro de los mecanismos de ETL para poder automatizar parte de éste proceso que actualmente se realiza parcialmente de manera manual, y además para poder garantizar una total integración de los datos, que no se consigue con el proceso ETL actual.

4.8.1. Kettle - Pentaho

La tecnología de Pentaho para la integración de datos, conocida como Kettle, es una herramienta de software libre que ofrece unos potentes mecanismos para la extracción, transformación y carga de datos, con una interfaz gráfica en la que se puede diseñar de manera fácil e intuitiva los procesos de transformación de datos. En la figura 4.4 se puede visualizar el proceso ETL actual del proyecto usando la herramienta de diseño “Spoon” que se encuentra integrada en Kettle. Cada uno de los elementos que se observan incluye las tareas de transformación requeridas para adaptar los datos al nuevo sistema de almacenamiento.

4.8.2. Mirth Connect

Mirth Connect ofrece una interfaz para estándares HL7 multiplataforma y de software libre que permite el envío bidireccional de mensajes HL7 entre sistemas y aplicaciones bajo una lincencia pública de Mozilla 1.1 (MPL 1.1).

Mirth Connect tiene una arquitectura basada en canales para conectar diversas tecnologías médicas (Health Information Tecnology) y permitir que los mensajes sean filtrados, transformados y dirigidos, basados en una serie de reglas altamente personalizables. Los canales consisten en conectores, tanto de entrada como de salida, en filtros y transformadores, que pueden ser encadenados unos con otros.

Los conectores pueden ser configuradores para aceptar conexiones de diferentes protocolos y medios. Los conectores fuentes pueden ser usados para designar un tipo de receptor para aceptar los mensajes recibidos a través de TCP/IP o a través de

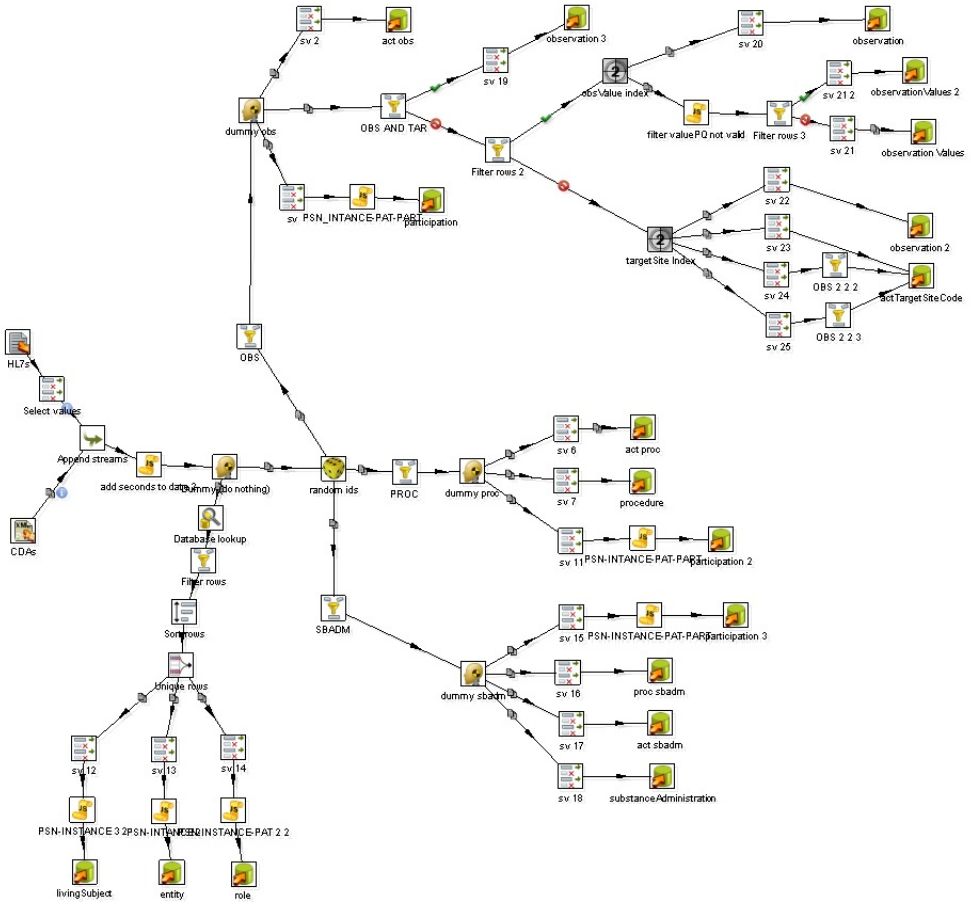

Figura 4.4: Proceso ETL usando herramienta Kettle

un web service. Los conectores de destino son usados para especificar el destino de los mensajes salientes, hacia una aplicación server o a una base de datos. Todos los mensajes y transacciones son opcionales y pueden ser registrados en una base de datos local. Mirth Connect ofrece un soporte para la conexión con multitud de protocolos, como: TCP/MLLP, texto plano, PDF y RTF, JMS, FTP/SFTP, HTTP, SMTP, SOAP, y con múltiples bases de datos entre las que se encuentra MySQL, PostgreSQL, Oracle, Microsfot SQL Server y ODBC.

Es justamente la gran variedad de configuración de conectores, junto con la posibilidad de ejecución scripts basado en una pequeña variación de javascript durante los procesos de transformación, lo que permiten a ésta tecnología integrar o integrarse con otras tecnologías. Facilitando así enormemente la integración del servicio TermBinding en ésta tecnología.

Capítulo 5

Especificación de requisitos

5.1. Introducción

Ésta capítulo reúne la Especificación de Requisitos Software(ERS) del servicio de enlace entre HL7 RIM y terminologías médicas(TermBinding). Las directrices seguidas para la elaboración del presente capítulo se encuentran en "IEEE Recommended Practice for Software Requirements Specifications" (IEE Estándar 830-1998)[21].

Se ha adoptado un formato para la identificación unívoca y posibilitar la trazabilidad de los requisitos de la manera que sigue:

REQ#XX: *Descripción en lenguaje natural.*

Adicionalmente, al final de éste capítulo se puede encontrar una sección dedicada a los casos de usos obtenidos a partir de los requisitos.

5.1.1. Propósito

El propósito de éste capítulo es establecer una lista forma y exhaustiva de los requisitos software para el sistema del presente proyecto. La ERS define todas las características, restricciones de diseño y funcionalidades que el sistema debe cumplir y servirá para validar la corrección e integridad del sistema.

El documento está orientado tanto a los desarrolladores como a los usuarios. Para conocer los requisitos que el sistema cumple en caso de querer ampliarlo u optimizarlo en el caso de los desarrolladores, o para que el usuario final sepa lo que puede esperar del sistema.

5.1.2. Ámbito del Sistema

La justificación del desarrollo del presente servicio radica en la necesidad de la total integración de los datos provenientes de diversos ensayos clínicos con un modelo de datos basados en el HL7 RIM. El servicio de enlace entre HL7 RIM y terminologías médicas, denominado TermBinding en el ámbito del proyecto, propone una solución para la total integración de estos datos, para permitir una libertad y mayor facilidad a la hora de acceder a esos datos. De ésta manera se permite a los usuarios que puedan realizar búsquedas por elementos no específicos, si no sobre generalidades, términos adyacentes o de elementos relacionados dentro de la terminología. Así se podrá integrar gran parte de la expresividad que ofrecen las terminologías médicas, y adaptarlo a nuestro propio modelo de datos, perdiendo la menor cantidad de información posible que se encuentra implícita en la expresividad de los lenguajes médicos.

El servicio por tanto, resolverá la asignación de un concepto dado de las terminologías para que pueda ser almacenado o recuperado de elementos del modelo de datos. La resolución de la asignación a elementos del modelo de datos también se realizará para los conceptos relacionados con el concepto original. Los elementos relaciones se obtendrán de la formalización del concepto, proceso en el cual se extraerán los pares de conceptos que conforman sus relaciones. Los pares de conceptos que conforman una relación, como ya se comentó en la pasada sección de la forma normal, están constituidos por el concepto que especifica el tipo relación y por el concepto con el que se encuentra relacionado, siendo éste último el que sea resuelto en el servicio del TermBinding. Se tendrá en cuenta el contexto en el que se encuentre el concepto a almacenar de una relación, así como el significado del tipo de relación con el que se encuentre vinculado al concepto original.

5.1.3. Visión general del documento

La ERS a definir a continuación se encuentra dividida en tres secciones.

- Introducción en la que se proporciona una visión general de la ERS. Especifica: propósito; ámbito del sistema; definiciones, abreviaturas y acrónimos; referencias; y la presente visión general del documento.

- Factores que afecta al producto y por tanto a la ERS, proporcionando una descripción general del sistema detallando las funciones que se llevarán a cabo por el sistema, así como sus restricciones, supuestos y dependencias que deben aplicarse al producto.

- Definición de los requisitos de manera detallada que debe satisfacer el sistema, proporcionando toda la información que se pueda a llegar a necesitar en su desarrollo. Se incluirán de manera adicional requisitos no funcionales como posibles interfaces externas, restricciones de rendimiento, de diseño o de seguridad.

5.1.4. Definiciones, Acrónimos y Abreviaturas

5.1.4.1. Acrónimos

A continuación, en la tabla 5.1, se definen los acrónimos utilizados en la especificación de requisitos software para facilitar la escritura y comprensión del capítulo.

Acrónimo	**Definición**
ERS	Especificación de Requisitos Software
CDM	Common Data Model
SPARQL	SPARQL Protocol and RDF Query Language
ETL	Extract Transform and Load
IHTSDO	International Health Terminology Standards Development Organization
SNOMED CT	Systematized Nomenclature of Medicine - Clinical Terms
LOINC	Logical Observation Identifiers Names and Codes
HGNC	HUGO Gene Nomenclature Committee
OWL	Ontology Web Language
API	Application Programming Interface
RAM	Random Access Memory
GB	Gigabyte
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
XML	eXtensible Markup Language
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
JVM	Java Virtual Machine

Cuadro 5.1: Acrónimos utilizados en la ERS

5.1.4.2. Abreviaturas

De manera adicional, en la tabla 5.2 se incluyen las abreviaturas usadas a lo largo de la sección de requisitos para una mayor legibilidad del capítulo.

Abreviatura	Definición
Std	Estándar
SO	Sistema Operativo

Cuadro 5.2: Abreviaturas utilizados en la ERS

5.1.5. Referencias

IEEE Recommended Practice for Software Requirements Specifications (IEEE Std.830-1998).

5.2. Descripción General

A continuación se realizará una descripción de alto nivel del sistema y se describirán aquellos factores que afecten al producto y a sus requisitos. Se describirá su contexto y se proporcionarán las pautas necesarias para poder comprender los requisitos descritos, facilitando así la lectura y comprensión del apartado de requisitos.

5.2.1. Perspectiva del Producto

El sistema desarrollado en el presente proyecto se encuentra incluido como uno de los servicios de un sistema mayor. El servicio se encuentra ubicado como una interfaz para acceder a los datos del modelo de datos como se puede observar en la figura 5.1.

El servicio es parte del denominado “Core Dataset” e interactúa con otros servicios externos como son el Query Builder Service y el Data Push Service como se puede observar en la figura 5.2. Con el servicio Query Builder Service se interaccionará para analizar y resolver los conceptos requeridos en la consulta a realizar al sistema global, para poder construir así la query SPARQL con la que el usuario solicitará la información al CDM. El servicio Data Push Service hará uso del TermBinding para poder asociar un concepto que se quiera almacenar con un objeto del CDM, además de toda la información implícita en el propio concepto extraída a través de su normalización. Actualmente, en el ámbito general del proyecto global en el que se encuentra ubicado el TermBinding, se pretende incluir dicho servicio dentro del propio proceso de ETL del proyecto global.

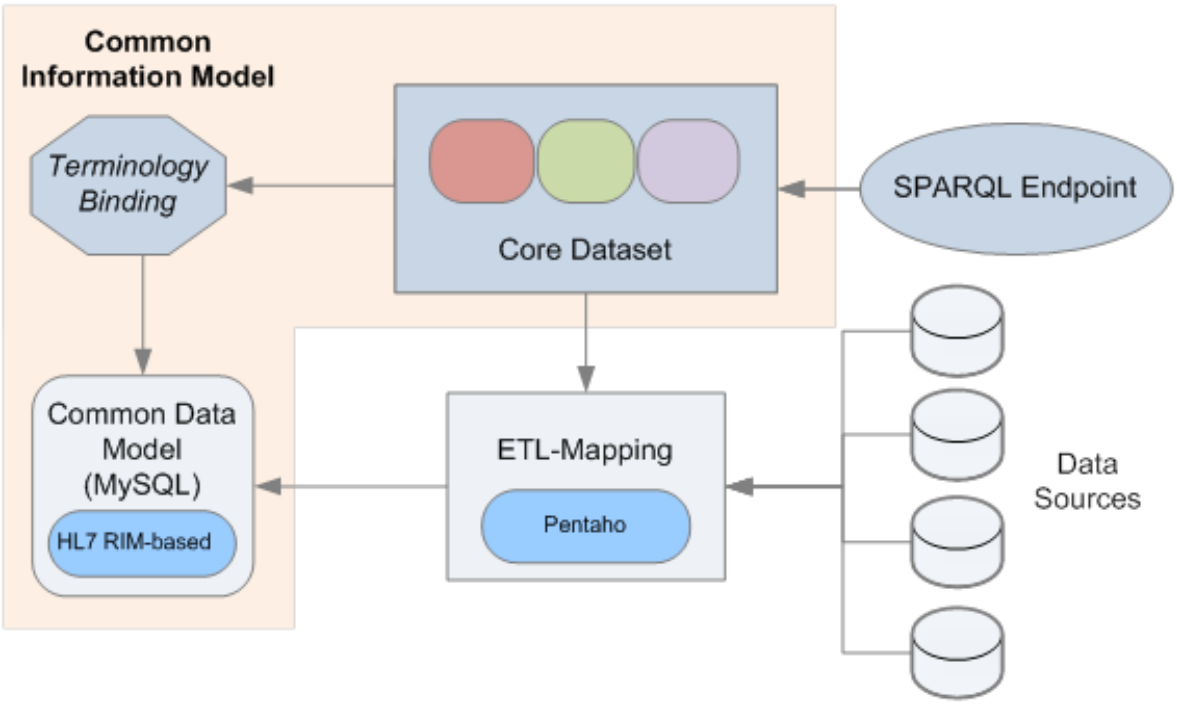

Figura 5.1: Esquema de TermBinding y resto de componentes

5.2.2. Funciones del Producto

La función única y principal del servicio es la de proporcionar un medio por el que indicar la asociación del contenido de una terminología médica, proporcionado por su código en dicha terminología y una cadena de caracteres que indica el concepto en sí, con elementos u objetos del modelo de datos. Aunque ésta función en sí se podría subdividir en tres enfoques diferentes dependiendo de las necesidades:

- Resolución de la clase del CDM a la que pertenece cierto concepto como unidad principal de información.
- Resolución del atributo al que pertenece un concepto que se encuentra contextualizado dentro de la semántica de un concepto principal.
- Resolución de los tipos de conceptos que pueden almacenarse dentro de un objeto del CDM. Es decir, el proceso inverso a los anteriores.

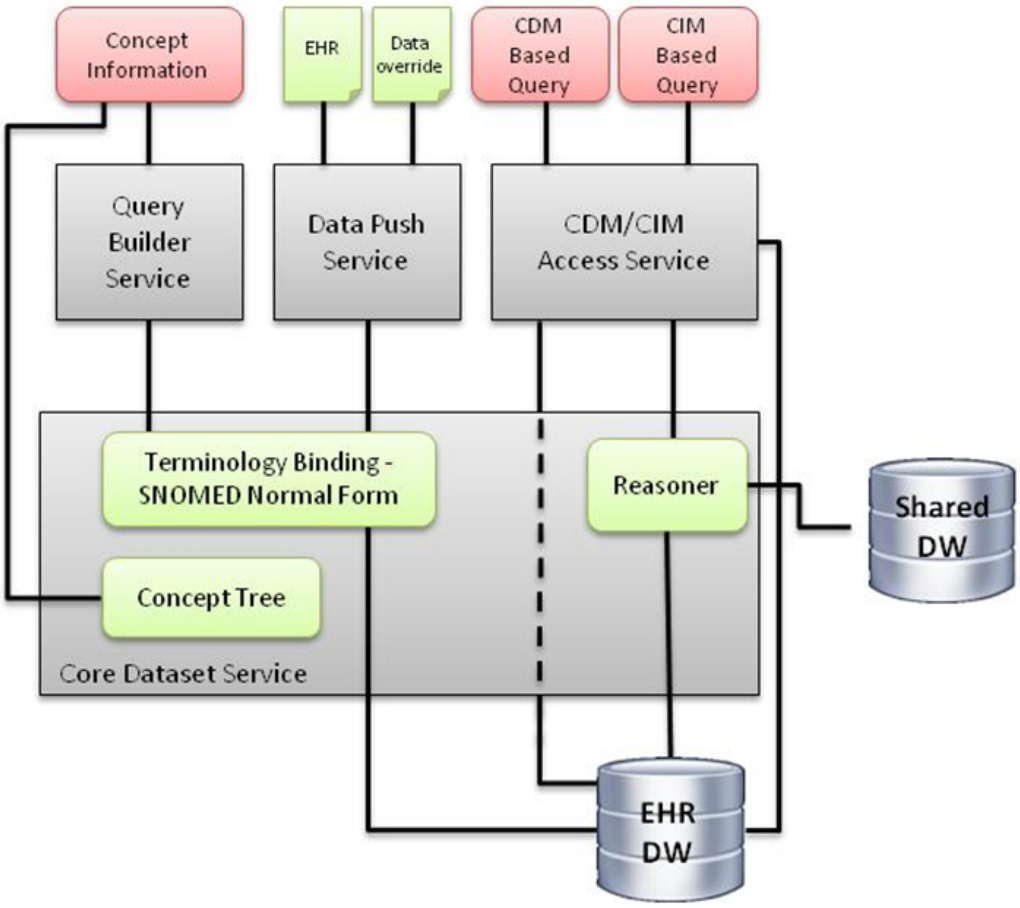

Figura 5.2: Esquema de interacciones TermBinding y resto de servicios

5.2.3. Características de los Usuarios

El servicio al formar parte de un sistema mayor, carecerá de usuarios físicos como tales, teniendo en cuenta un único rol de administrador de servicio, requerido para las labores de gestión de los recursos requeridos por el servicio. Ésta gestión se efectuará sobre los elementos relacionados a la generación de la ontología y las modificaciones que se deben realizar sobre ella para que la labor del servicio pueda tener lugar.

El **administrador** será el usuario encargado de actualizar la ontología del servicio, que nutre al "Core Dataset", con las actualizaciones de las terminologías médicas, que en el caso de SNOMED las realiza la IHTSDO cada aproximadamente 6 meses. Las actualizaciones de la ontología a usar por el servicio, se generarán a través de la ejecución de un script en Perl, que unirá las ontologías de las termino-

logías (SNOMED, LOINC y HGNC) junto con la información de las asignaciones de sus elementos al CDM. Debido a esto el administrador deberá tener unos mínimos conocimientos de Perl y con el lenguaje de definición de ontologías OWL, y estar familiarizado con la estructura de las publicaciones de las terminologías médicos, sobre todo para el caso de SNOMED CT. Para una mejor labor de la administración, éste deberá poseer algunos conocimientos del framework Sesame, que es donde se desplegarán finalmente la ontología creada.

El **usuario** del servicio serán el resto de servicios del proyecto en el que se encuadra, y visualizarán la información que es extraída por el TermBinding mediante la disposición de una interfaz al usuario. El usuario, para poder interpretar y dar uso de ésta información, deberá tener unos mínimos conocimientos clínicos, así como de las terminologías médicas en los que la información clínica puede ser expresada (específicamente de SNOMED CT, LOINC y HGNC).

5.2.4. Restricciones

A pesar de que el servicio se encuentre incluido como parte de un producto mayor, el servicio puede desplegarse de manera independiente y ser resuelto y accedido por el resto de servicios desde internet.

El servicio en sí carece de restricciones de SO, ya que el servicio en sí requiere ser deplegado en un contenedor Apache Tomcat, el cual puede ser ejecutado tanto en SO basados en windows como en Linux, ocurriendo lo mismo con Sesame ya que se encuentra basado en Java.

Se necesitará por tanto una máquina virtual Java SE Runtime Environment 6 o superior así como las API requeridas de openRDF de Sesame para Java para el manejo de ontologías.

Para el despliegue del servicio se necesitará de una versión de Apache Tomcat 6.0 o superior para que pueda ser accesible por el resto de elementos del sistema en el que se encuentra ubicado.

De manera adicional, para contener la ontología de las terminologías, se podría dar uso de la API para Java de Sesame o bien se podría dar uso del framework en sí.

Una de las limitaciones a tener en cuenta durante el desarrollo y disposición del servicio son las necesidades hardware que se requieren para el montaje de la ontología que unifica todas las terminologías. Para poder desplegar en Sesame dicha ontología se requiere de aproximadamente 3GB de RAM. Ésta limitación es algo muy a tener en cuenta para que el servicio pueda ejecutarse con normalidad para que se pueda ofrecer un servicio con un rendimiento aceptable.

5.2.5. Suposiciones y Dependencias

A continuación se tratará de describir los factores que en caso de cambio puedan afectar a los requisitos así como de las dependencias existentes en el sistema que puedan verse afectadas por cambios en otras partes del sistema.

5.2.5.1. Suposiciones

Se supone que los requisitos a especificar serán definitivos e inalterables, pero en caso de cambio en los mismos se realizará un procedimiento controlado y exhaustivo, reflejándose en los documentos oportunos.

5.2.5.2. Dependencias

La principal dependencia existente en el actual estado del servicio se encuentra en las actualizaciones que se realizan en las terminologías, que en el caso de SNOMED CT se realiza de manera semestral. Estos cambios que se realizan en las terminologías son realizados de manera externa al servicio, para en teoría, una mejor definición del vocabulario que representan dichas terminologías. Lo cual supone que términos ya asignados a objetos del modelo de datos pueden verse afectados por dichos cambios en los vocabularios.

Debido a que parte de las consultas a realizar a la ontología depende de la estructura y organización de la misma, las consultas que se realizan de manera sistemática en la parte programática del servicio pueden verse afectas por cambios que se realicen en la ontología. Por ello, hay que tener en cuenta la dependencia existente entre el código del servicio y la ontología que se está haciendo uso, ya que cualquier cambio en su estructura requiere de cambios en las consultas predefinidas dentro del código del servicio.

De manera adicional, existe una dependencia de la API usada para el desarrollo del servicio. El servicio basa parte de su desarrollo en el uso de la API de OpenRDF para la gestión de la ontología creada. Ésta dependencia en caso de cambio de API, o de problema de versiones, supondría la reescritura del parte del código del servicio.

5.3. Requisitos Específicos

En ésta sección se exponen los requisitos que satisfarán las necesidades del servicio. Los requisitos aquí presentados se han especificado teniendo en cuenta el criterio de “testeabilidad”, es decir, dado un requisito, debe ser demostrable si es satisfecho o no.

Comentar inicialmente, que al tratarse de un servicio que forma parte de un producto mayor, los requisitos específicos serán los concernientes a la comunicación al resto de componentes, así como los requisitos funcionales.

5.3.1. Interfaces Externas

A continuación se describen los requisitos que afectan a la interfaz con otros sistemas e interfaces de comunicaciones. Se omitirán los requisitos de la interfaz de usuario al carecer de dicha interfaz al usuario, al ser el presente servicio desarrollado parte de un sistema mayor.

5.3.1.1. Interfaces Hardware

- **REQ#01:** El sistema deberá funcionar sobre diversas plataformas o arquitecturas hardware como x86, x64, etc.

5.3.1.2. Interfaces Software

- **REQ#02:** El sistema deberá funcionar sobre diversas plataformas software como Windows, Linux y Solaris.

5.3.1.3. Interfaces de Comunicación

- **REQ#03:** El sistema utilizará los protocolos de comunicación por Internet TCP/IP, HTTP y HTTPS.
- **REQ#04:** El sistema utilizará el estándar XML y el protocolo SOAP para el intercambio de mensajes entre y con el resto de servicios.

5.3.2. Funciones

Los requisitos funcionales especifican las acciones que deberán llevarse a cabo por el sistema. Se listarán los requisitos funcionales siguiendo los objetivos a alcanzar por el servicio. Por cada objetivo se determinara la entrada y el resultado a obtener, y además por cada objetivo se detalla las funciones que permiten llevarlo a cabo. Se ha elegido éste criterio de ordenación ya que los requisitos del sistema están estrechamente relacionados por los servicios que debe de ofrecer al resto de componentes.

- **REQ#04:** El sistema aceptará códigos de la terminología SNOMED CT.
- **REQ#05:** El sistema aceptará códigos de la terminología LOINC.

- **REQ#06:** El sistema aceptará códigos de la terminología HGNC.
- **REQ#07:** El sistema aceptará por cada petición uno o más códigos.
- **REQ#08:** El sistema aceptará de manera indiferente códigos que puedan pertenecer a diferentes terminologías.
- **REQ#09:** El sistema devolverá la tabla del CDM en la que debe almacenarse el concepto solicitado.
- **REQ#10:** El sistema extraerá la forma normal de SNOMED CT para códigos de dicha terminología. Los conceptos que se extraigan de dicho proceso tendrá que ser asignados a objetos del CDM.

5.3.3. Requisitos de Rendimiento

En ésta sección se detallan los requisitos relacionados con la carga que se espera que tenga que soportar el sistema. Se describirán de manera numérica y cuantificable. También se especifican los requisitos de los datos, que afecten a la información que se vaya a intercambiar en el sistema.

Los requisitos de rendimiento del servicio son particularmente relevantes al verse involucrados procesos de razonamiento semántico que pueden llevar un tiempo relativamente alto. Por ello, toda petición realizada al servicio debe ser procesada de la manera mas eficiente posible, incidiendo sobre todo en los tiempos de ejecución de las consultas SPARQL que se realizan dentro de la ejucución del servicio como parte del razonamiento.

5.3.4. Restricciones de Diseño

En ésta sección se especifica todo lo que pueda restringir las decisiones relativas al diseño del sistema, como estándares, limitaciones de hardware, etc.

- **REQ#11:** El sistema usará tecnologías de páginas Web dinámicas.
- **REQ#12:** El sistema se ajustará al estándar SOAP para el intercambio de datos. Siendo el principal interés el diseño de las estructuras de datos usadas para el intercambio de datos con otros servicios, es decir, que el diseño de las estructuras de datos del servicio sean compatibles con el estándar SOAP.
- **REQ#13:** El sistema se deberá aprovechar los recursos y plataformas ya existentes para el razonamiento semántico. Como son la plataforma Sesame.

5.3.5. Atributos del Sistema

En ésta sección se detallan los atributos de calidad del sistema como son: Fiabilidad, mantenibilidad, portabilidad y la seguridad. Se especificará qué tipo de usuario están autorizados o no, y cómo se implementarán los mecanismos de seguridad.

5.3.5.1. Seguridad

Para la seguridad del servicio se usará el paquete javax.net.ssl. Dicho paquete provee de las clases e interfaces necesarias para usar el protocolo SSL y el sucesor TLS. Ésta API permite tanto al cliente como al servidor la selección de la versión SSL o TLS a usar, así como el sistema de cifrado que más les convenga.

Para implementar dicho sistema de seguridad en nuestro servicio se tendrá que adquirir la keyStore y la contraseña, y establecerlas como propiedades del sistema para que el servicio pueda comunicarse o solicitar el servicio de otros elementos del sistema.

No existirán mas mecanismos de seguridad que tenga que implementar el presente servicio, ya que el la gestión de usuarios y permisos de los mismos es implementada en otra parte del proyecto en el que se encuentra ubicado el presente servicio.

5.3.5.2. Fiabilidad

La fiabilidad del servicio está supeditada a un riguroso trabajo de investigación sobre una cantidad extensa de datos procedentes de terminologías, y esto puede provocar que tras la adquisición de nuevos conjuntos de información, puede ocurrir que dichos conjuntos de datos incluyan términos o conceptos, o que dichos conceptos posean atributos aún no totalmente integrados. Esto implica que se deberá vigilar los nuevos conjuntos de datos a integrar en el sistema para velar por la fiabilidad del servicio y los datos a almacenar en el CDM.

Independientemente de la compleja labor de unión de datos entre conceptos de terminologías y objetos del modelo de datos, la fiabilidad del sistema como servicio independiente tiene un funcionamiento correcto, habiendo sido analizado para consultas de tamaños muy grandes. Por tanto tan sólo se dependerá de la fiabilidad de elementos externos como conexión a internet y tráfico existente, y la carga del servidor en el que se encuentre contenido la plataforma Sesame con la ontología correspondiente.

5.3.5.3. Portabilidad

El servicio desarrollado puede ser completamente independiente del resto de servicios, y además puede ser ejecutado sobre cualquier sistema operativo o plataforma hardware en la que se pueda instalar una JVM y Apache Tomcat.

5.3.5.4. Mantenibilidad

La mantenibilidad del sistema queda relegada a la actualización del trabajo de integración de datos así como de las actualizaciones periódicas de las terminologías de las que depende el servicio para el razonamiento semántico. El mantenimiento podría ser omisible en un sistema para un uso normal, ya que las actualizaciones de las terminologías son una constante para el perfeccionamiento de dichos vocabularios, ya que en dichas actualizaciones se incluyen nuevos términos, se revisan los ya incluidos, y en la mayor parte de los casos, gran parte de las modificaciones de una versión a otra son cambios en las etiquetas de los conceptos para una más clara visualización y comprensión del concepto definido.

Capítulo 6

Diseño e implementación del sistema

El diseño y la implementación del servicio de la solución propuesta se describirá en el presente capítulo. El diseño reunirá los diagramas que describan el diseño lógico del sistema, incluyendo el diagrama de clases software y los diagramas de interacción.

6.1. Diagrama de secuencia UML

El diagrama de secuencia es un tipo de diagrama usado para modelar la interacción existente entre objetos de un sistema según UML. Un diagrama de secuencia muestra la interacción del conjunto de objetos en una aplicación a través del tiempo. Se ha elegido éste modelo como diagrama de secuencia para mostrar el diseño del servicio debido a que facilitan la visualización de la secuencia de mensajes.

En la figura 6.1 podemos observar las interacciones que se producen al recibir la petición sobre un concepto cualquiera. Se han incluido las principales interacciones que se producen y omitido las de menor relevancia. Las interacciones con servicios externos como NormalFormGenerator con otros componentes también han sido omitidos por motivo de mantener claridad.

6.2. Diagrama de clases

Con la finalidad de describir la estructura del sistema, mostrando sus clases, atributos, y relaciones entre ellos. A continuación se presentará por tanto el diagrama general de clases, cuya descripción detallada será objeto de la sección posterior.

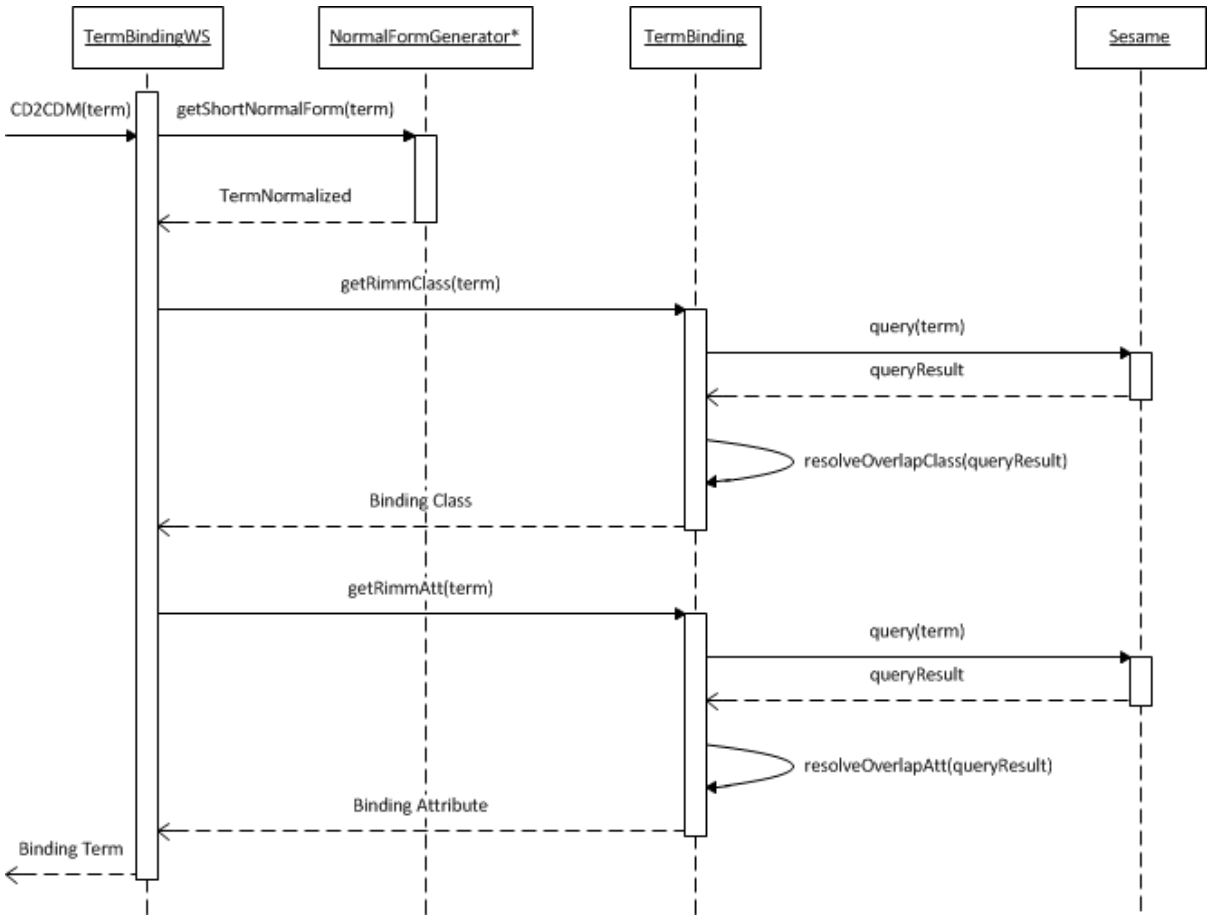

Figura 6.1: Diagrama UML de interacción

Como se puede apreciar en la figura 6.2 existen dos clases principales que contienen la parte funcional del sistema, y una clase adicional que es la que permite generar y enriquecer la ontología que contiene las diferentes terminologías, y de cuyo conocimiento se nutre el sistema.

La clase TermBindingWS sería la que se desplegaría como servicio, y ofrecería los métodos principales del sistema, y como se apreció en el diagrama de secuencia será que se comunique con la clase externa para la normalización de conceptos SNOMED CT, y con la clase del sistema TermBinding para la resolución de los conceptos de las terminologías a elementos del CDM.

La clase TermBindingLoader es una clase creada como uno de los pasos necesarios para poder realimentar la ontología de las terminologías con información de los "bindings" o uniones de las terminologías con un modelo de datos basado en HL7 RIM.

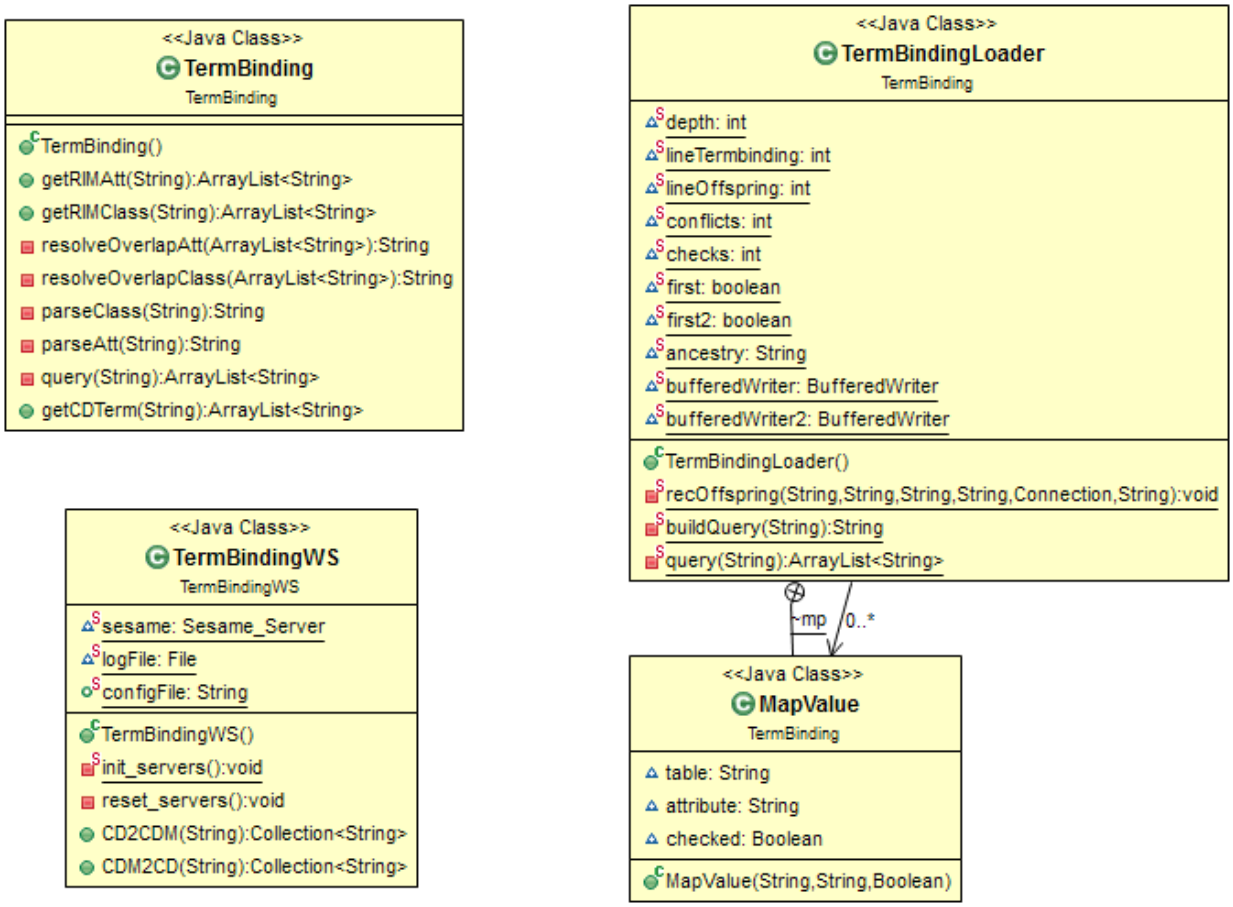

Figura 6.2: Diagrama de clases

6.3. Descripción de Clases Software

Para tener un mayor entendimiento de las clases, en ésta sección se detallarán los aspectos más importantes y relevantes de las clases del sistema. Se incluirá además de la descripción de las mencionadas clases, las especificaciones de los atributos y métodos más representativos de éstas.

6.3.1. Termbinding Web Service

La clase TermBindingWS principalmente ofrece la interfaz necesaria para acceder al sistema mediante su despliegue como servicio web en el contenedor Apache Tomcat.

Los métodos a publicar en el correspondiente WSDL serán los métodos principa-

les requeridos para el sistema CD2CDM(String) y CDM2CD(String). Estos métodos serán accesibles como recursos web para la resolución de conceptos, y devolverán las clases y atributos del modelo de datos a la que pertenecen, así como las asignaciones de clases y atributos de los conceptos que se extraigan de la normalización SNOMED CT del concepto, si es que existen. Ésta parte de la lógica del servicio se debe de implementar en ésta clase, ya que es la que se produce la normalización SNOMED CT de un concepto de dicha terminología. Esto es debido a que parte de la clasificación de un concepto requiere conocer el contexto en el que se encuentra un concepto.

La clase también posee unos métodos para el control del framework Sesame, que forman parte del arranque del servicio en si. Estos métodos son necesarios ya que el framework sesame es requerido no solo por el sistema TermBinding en sí, si no también es requerido por la clase externa NormalFormGenerator, no incluida en éstas descripciones, pero si mencionada en el diagrama de secuencia.

Los atributos de ésta clase son configuraciones del servicio y del servidor Sesame, así como la especificación de archivos de registro para proporcionar información adicional del servicio o feedback.

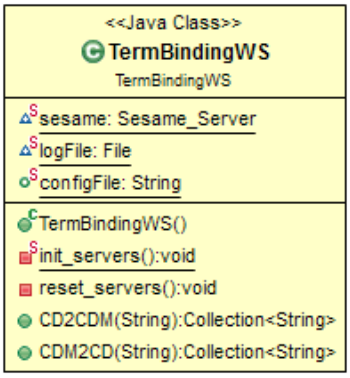

Figura 6.3: Clase TermBindingWS

6.3.2. TermBinding

La case TermBinding es la que realiza casi la totalidad de la lógica necesaria para resolver la asignación de un concepto a un elemento del modelo de datos. Ésta clase usará la información incluida de manera adicional en la ontología para poder determinar la pertenencia de una un concepto a una clase u otra del CDM, o la pertenencia de un concepto como atributo del CDM de otro concepto incluido como clase.

Los métodos principales de la clase sería el que resuelven la asignación de un concepto a una clase del CDM, getRIMClass(String), y el método que resuelve la asignación de un concepto a un atributo del CDM, getRIMAtt(String). Estos conceptos dan uso una serie de conceptos auxiliares para poder llevar a cabo su labor.

Dentro de los métodos auxiliares, existen dos métodos para la resolución de los solapamientos que se producen en la interoperabilidad entre vocabularios médicos y los modelos de datos basados en el HL7 RIM.

Finalmente, los métodos que proporcionan un apoyo o interfaz para la comunicación entre el sistema y el Sesame server. El método query(String) será el que envíe la query a la plataforma Sesame para que sea ejecutada contra la ontología, y cuyos resultados serán devueltos y procesos por los métodos que extraigan la información requerida por los métodos principales.

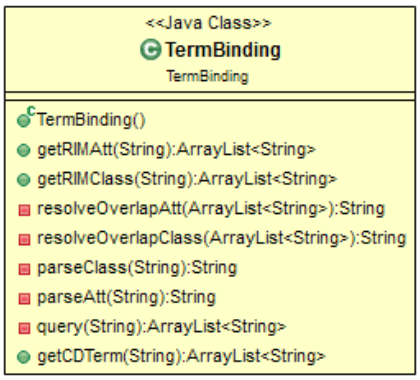

Figura 6.4: Clase TermBinding

6.3.3. TermBindingLoader

La clase TermBindingLoader sería una clase diseñada para facilitar la generación y modificación de la ontología base de SNOMED CT, para añadirle lo requerido para poder realizar razonamiento semántico para la asignación de elementos del vocabulario médico con objetos del modelo de datos CDM. Ésta clase recorre y anota la clasificación de todos los conceptos de todas las terminologías, produciendo un archivo con toda ésta información. Dicho archivo es el utilizado para poder generar la ontología a través de un script en perl que genera el OWL que integra todas las terminologías, así como su clasificación. Dicho OWL es el que posteriormente se almacena en un repositorio semántico SESAME sobre el que se pueda realizar el razonamiento semántico requerido por la clase "TermBinding".

El método principal de éste clase es directamente el de la ejecución de la clase, desde donde se acceden al resto de métodos. El método recOffspring es un método al

que se accede de manera recursiva para acceder a todos los nodos de la terminología SNOMED CT. El grafo que se puede deducir de la estructura de la terminología SNOMED CT es la de un grafo acíclico, por lo que para recorrer su totalidad se establece la clase MapValue para marcar cada nodo como visitado o no, y evitar profundizar por una rama ya visitada. Esto tiene una excepción, ya que partes del árbol de SNOMED CT pueden pertenecer simultáneamente a clases diferentes del CDM (formando lo denominado como solapamiento), el cual es resuelto en la lógica del servicio TermBinding. Debido a estos solapamientos, si estamos visitando un subárbol que ya ha sido recorrido y se encuentra marcada como recorrido, se tendría que recorrer de nuevo siempre y cuando se esté accediendo a él para asignarlo a otro objeto del CDM.

El resto de métodos y atributos son herramientas para facilitar las consultas al repositorio Sesame, método Query(String), o en el caso de muchos de los atributos, para la extracción de información estadística sobre el número de nodos recorridos, nodos solapados etc.

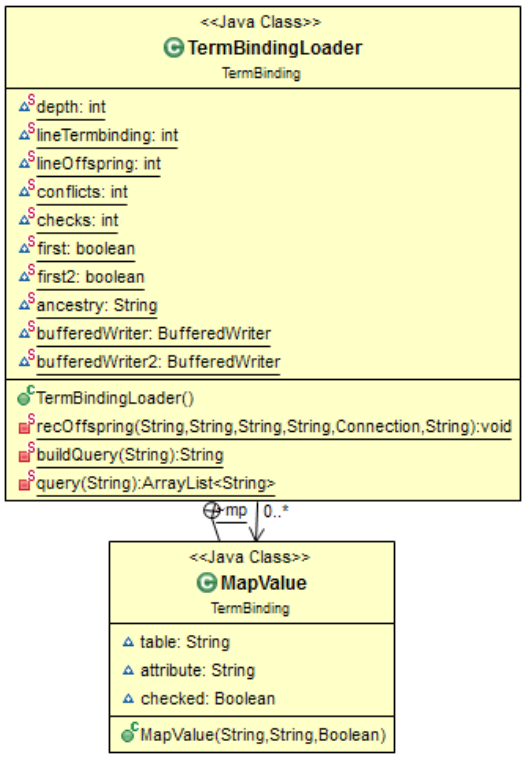

Figura 6.5: Clase TermBindingLoader

Capítulo 7

Resultados y pruebas

En el actual capítulo se pretende mostrar los resultados que se han obtenido de los métodos hasta ahora explicados, y las pruebas a las que se ha sometido para mostrar su respuesta ante los diferentes esfuerzos a los que puede ser sometido. Los resultados que se extraigan serán analizados e interpretados, a pesar de que durante la misma elaboración de la herramienta ya se han ido realizando pruebas para el correcto funcionamiento del servicio, pero como en todo sistema, hay pruebas que no se realizan hasta llegados el actual punto de completitud del servicio.

Debido a la simplicidad programática del servicio en sí, no se ve necesario la realización de pruebas unitarias e integración del servicio de manera específica para cada una de ellas. Por ello, se realizarán unas pruebas globales de rendimiento del servicio.

7.1. Análisis de los resultados

Usando los métodos hasta ahora explicados para el desarrollo del servicio, se han extraído una serie de resultados a partir de los datos que actualmente se encuentran integrados. El análisis de estos resultados pretende mostrar una visión general de la integración de los datos, así como las cuestiones encontradas resultantes de aplicar los métodos hasta ahora descritos.

7.1.1. Distribución de los datos integrados

Con la solución propuesta se ha conseguido integrar los datos de ensayos clínicos realizados a mas de cincuenta pacientes con un total de 2400 actos realizados. Como un balance general de la distribución de estos datos, comparada con la distribución de todos los conceptos de SNOMEC CT, LOINC y HGNC entre las clases principales

del modelo de datos, cabe destacar una serie de correlaciones existentes en ésta distribución.

La mayor parte de los datos integrados de estos 2400 actos, corresponden con observaciones realizadas a los pacientes (más de la mitad de los actos), lo cual coincide con la distribución general de las terminologías entre las clases del modelo de datos. Esto tiene principalmente dos motivos:

- Todo concepto de la terminología LOINC se trata de observaciones realizadas, algo normal teniendo en cuenta el fin para el que ha sido desarrollado dicha terminología.
- La mayor de los conceptos pertenecientes a SNOMED CT (cercano a la mitad de ellos) son conceptos que se pueden clasificar como observaciones dentro del modelo de datos.

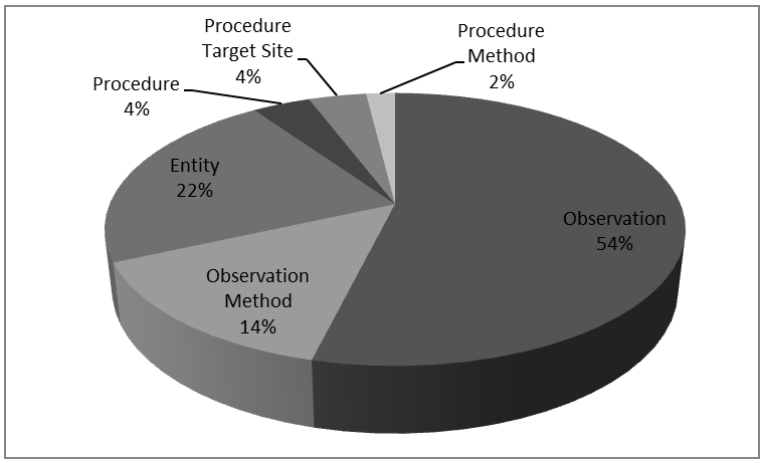

Figura 7.1: Distribución de los datos actualmente disponibles

Como se puede apreciar en la figura 7.1, el mayor volumen informativo de los datos integrados corresponde con observaciones realizadas a los pacientes (suma de los datos de "Observation" y "Observation Method").

Por otro lado nos encontramos con un número elevado de "Entity", cabiendo comentar que dentro de esta clasificación como entidad, se encuentran tanto conceptos utilizados para expresar administraciones de sustancias, siendo las sustancias en sí consideradas entidades, como entidades que pueden formar parte de procedimientos y observaciones (ej. el trocar utilizado en una biopsia realizada con un trocar).

Finalmente, se puede apreciar que a pesar de que la menor cantidad de los datos son los que representan los procedimientos realizados a los pacientes, son estos procedimientos los que en su integración se extrae un mayor volumen de información

por cada concepto integrado. Ya que al contrario que las observaciones, en la mayor parte de los procedimientos expresados en SNOMED CT, viene implícito multitud de información. Esto se debe a que en la mayor parte de los procedimientos se puede expresar una mayor cantidad de información que indique las técnicas empleadas ("Method"), el lugar donde se practica el procedimiento ("Target Site"), instrumental utilizado ("Entity"), y vías de aproximación al lugar que tiene por objeto el procedimiento ("Approach Site"). Gracias a ésta mayor complejidad de los conceptos relacionados con procedimientos, se puede verificar el buen funcionamiento del servicio en la clasificación de la información extraída de la normalización de conceptos SNOMED CT.

7.1.2. Cuestiones encontradas

Una de las mayores cuestiones encontradas en la elaboración del proyecto, se debe a una situación en la que un concepto de SNOMED CT puede ubicarse en diferentes sitios dentro del modelo de datos. Ésta cuestión es advertida por el proyecto TermInfo 2, pero a pesar de que es advertida, no se ofrece ningún tipo de solución.

Analizando la terminología de SNOMED CT, nos encontramos con que existen más de 20000 conceptos solapados, o que su clasificación se encontraría con el problema de que dicho concepto puede ubicarse en dos lugares diferentes del "Common Data Model". Analizando dicha situación, se extrae que la mayor parte de los solapamientos se producen entre conceptos de pertenecen simultáneamente a "Procedure" y a "Observation", siendo minoría el resto de solapamientos, entre los que cabe destacar los producidos entre conceptos que pueden considerarse "Procedemiento" y "Substance Administration".

Para poder explicar esta situación, analizaremos la causa de uno de estos solapamientos. La estructura de SNOMED CT coincide con la de un grafo acíclico dirigido. Esto significa, en el ámbito de SNOMED CT, que a pesar de que sigue la estructura de un árbol, siendo cada concepto un nodo de dicho árbol, un nodo puede tener múltiples padres. Esto conlleva que si dos conceptos encontrados a menor profundidad, son clasificados en clases diferentes del modelo de datos, puede ocurrir que tengan parte de la descendencia en común, produciendo el denominado solapamiento.

Como se puede observar en la figura 7.2, el concepto "Procedure" tiene como uno de sus descendientes el concepto "Evaluation Procedure". Como se puede suponer, el concepto "Procedure" se clasifica como procedimiento, y todos los conceptos que deriven de él, deberán de ser ubicados en la respectiva clase "Procedure" del modelo de datos. Pero en cambio nos encontramos que todos los conceptos que sean "Evaluation Procedure" se consideran observaciones a efectos prácticos, por lo que todo concepto que derive de él, deberá de ser almacenado como una observación.

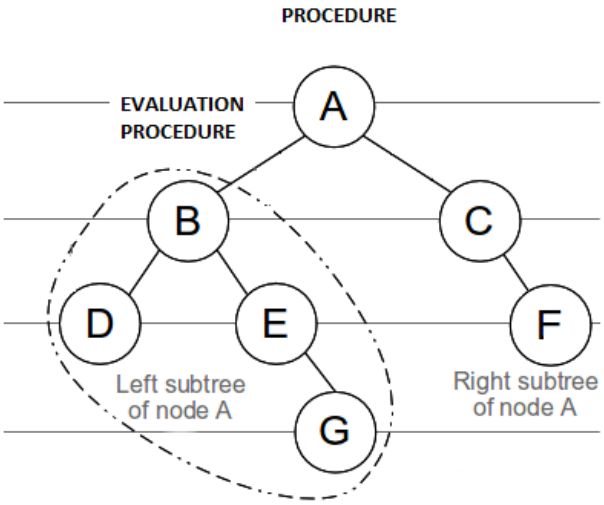

Figura 7.2: Ejemplo de solapamiento en la estructura de SNOMED CT

Como se puede esperar, esta situación provoca un solapamiento en la clasificación de los conceptos descendientes de "Evaluation Procedure", pudiéndose clasificar como observaciones o procedimientos. En éste caso concreto, se puede adoptar una solución en la que como nos encontramos con un caso en el que una generalidad de conceptos deben ser clasificados de una manera, pero un grupo pequeño de ellos, deben de ser clasificados de otra manera, por lo que para resolver dicho solapamiento, simplemente habría que considerar ésta excepción.

Los solapamientos entre conceptos de SNOMED CT, no son en su totalidad producidos por lo explicado hasta hora, ya que existen multitud de casos específicos en los que se produce ésta cuestión. Como la diferenciación entre conceptos que pueden ser métodos de un procedimiento y una observación, o métodos que pueden considerarse un procedimiento en sí. Debido a la multitud de causas que producen solapamientos, sólo se considera explicar la causa del mayor número de ellos (mostrado en la figura 7.2) y mencionar otros ejemplos de ellos sin entrar en detalle.

7.2. Pruebas de rendimiento

Las pruebas a realizar serán de sometimiento de esfuerzo del servicio en sí, mostrando su posible comportamiento ante la exposición de un número elevado de peticiones. Las pruebas de rendimiento a realizar se enfocarán desde diferentes tipos de tests. Se desplegarán una serie de clientes de servicios, y dichos clientes simularán la consulta al servicio por conceptos de diferentes terminologías. Las peticiones se

realizarán desde una sola petición por usuario, pero múltiples usuarios, hasta múltiples peticiones por usuario, pero pocos usuarios. De ésta manera se analizará el comportamiento del servicio ante el stress que supone un alto número de peticiones de un usuario, o el comportamiento ante un requerimiento del servicio por parte de muchos usuarios.

Las pruebas se realizarán con el servicio desplegado en una máquina de las siguientes características:

- **Modelo CPU:** Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz
- **CPU:** 1995.269MHz 64 bits
- **Tamaño caché:** 20M
- **Memoria RAM:** 8GB
- **SWAP:** 10GB
- **Red:** 100Mb/s

De ésta manera, las pruebas que se realizarán sobre éste sistema consistirán, como se ha explicado, en requerir el servicio de dos maneras diferentes. Inicialmente ejecutaremos una serie de llamadas en secuencial al servicio, por parte de un solo cliente, que realizará desde una sola petición, hasta múltiples peticiones. Se apuntarán los tiempos que tome el servicio en ejecutar y resolver todas las peticiones. Después se evaluará el comportamiento del servicio ante su requerimiento desde un solo usuario con una sola petición, hasta múltiples usuarios con una sola petición, es decir, en paralelo. Se apuntarán los tiempos requeridos por el servicio desde el momento que el primer usuario realice su petición, hasta que la última petición requerida sea resuelta.

Los resultados que se extraen de las pruebas realizadas se pueden observar representados en la figura 7.3. Todas las peticiones realizadas sobre procesadas con éxito y de manera correcta, verificando los resultados de todas ellas son los que debieran.

Se puede observar en la figura 7.3, que según se incrementa el número de peticiones, realizadas por un sólo usuario o bien distribuidas por usuarios, el tiempo requerido para procesar todas las peticiones sigue una progresión constante en ambos casos. A pesar de que dicha progresión es constante, indica que al menos, el tiempo requerido según se aumenta el número de peticiones hasta el límite de las 400 peticiones, distribuidas o no, no se incrementa exponencial hasta el punto de la saturación, ni nada cercano a ello. Por lo que se puede deducir que el servicio gestiona correctamente una carga elevada hasta unos puntos que se consideran fuera

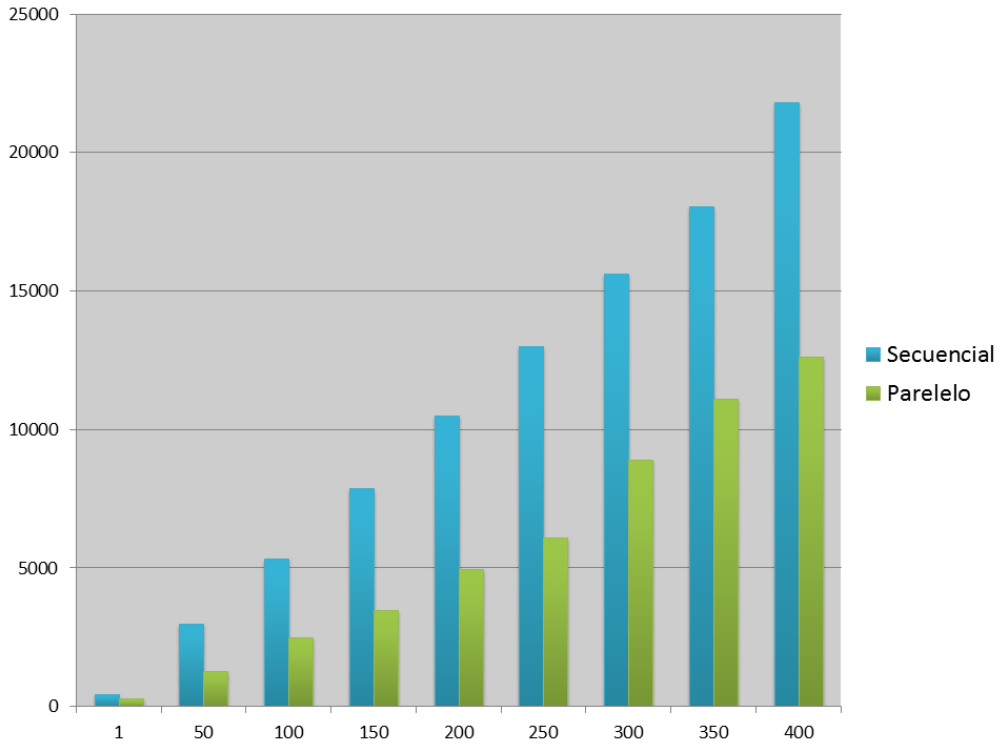

Figura 7.3: Rendimiento del Servicio Peticiones/ms

del uso habitual del servicio, como son las pruebas con hasta 400 peticiones. Por lo que se puede concluir una buena respuesta global del servicio.

Por otro lado, lo más evidente que se puede ver en los resultados, es la diferencia del tiempo requerido para procesar un mismo número de peticiones cuando éstas son emitidas en paralelo por múltiples usuarios o de manera secuencial por un solo usuario. Se puede observar que cuando las peticiones se distribuyen entre una cantidad idéntica de usuarios, el tiempo requerido es notablemente inferior al requerido si es realizado de manera secuencial. Ésta diferencia de tiempos es debida a como se realizan las peticiones por parte del cliente y como se procesan en el servidor. Si bien la parte servidora replica su parte ante múltiples requerimientos del servicio, debido a la buena elección de la tecnología Apache Tomcat para contener el servicio, ya que lo realiza de manera automática, reflejándose finalmente en el rendimiento en sí del servicio y en éstas pruebas realizadas. Por el contrario, si desde el punto de vista del cliente de pruebas, el cual intenta emular como pueden ser los usos que se harán del servicio, las peticiones se efectúan de manera secuencial y cada petición tiene que esperar a que la anterior sea resuelta, conlleva un peor rendimiento comparado con un cliente que distribuye las peticiones.

Para paliar éste posible defecto, que depende mayormente de la parte cliente más que de la parte servidora del servicio desarrollado, existen alternativas. Para mejorar el rendimiento secuencial y poder equipararse al rendimiento obtenido en el procesamiento de las peticiones en paralelo, y que esto solo suponga cambios en la parte servidora y permitir así al cliente olvidarse por completo de éste pequeño aspecto de rendimiento, el cambio a realizar sería modificar el parámetro de entrada del servicio de un único concepto a múltiples conceptos en una sola petición. Esto se deduce a raíz de que la mayor parte del tiempo invertido en el procesamiento de las peticiones se encuentra en el transporte y gestión de dicha petición, y no en su procesamiento ya en el servidor. Si se ofrece por parte del servidor el poder agrupar varias peticiones en una sola, permitiendo la consulta de múltiples conceptos en una sola petición, consigue rebajar la diferencia de tiempos obtenidos comparándola con la prueba paralela.

Capítulo 8

Conclusiones, líneas futuras y agradecimientos

Para finalizar la presente memoria, se exponen a continuación los logros conseguidos con el presente proyecto, una serie de conclusiones personales sobre el desarrollo del proyecto, y finalmente, unas posibles mejoras del servicio desarrollado.

8.1. Conclusiones

El servicio desarrollado y presentado en éste Trabajo Fin de Grado se ha planteado una solución a la propuesta para la creación de un servicio de enlace entre HL7 RIM y las terminologías médicas SNOMED CT, LOINC y HGNC. El presente proyecto da solución a dicha propuesta, pero sin embargo, la interoperabilidad semántica entre un modelo de datos basado en HL7 RIM y SNOMED CT, es algo en lo que actualmente hay grupos y proyectos de investigación intentando conseguir una interoperabilidad cada vez más completa. Esto implica que la solución propuesta consigue la interoperabilidad e integración de datos expresados con SNOMED CT y otras terminologías, para el ámbito de datos manejados en los proyectos INTEGRATE y EURECA.

A pesar de ésta puntualización, los objetivos planteados en el proyecto han sido resueltos satisfactoriamente en su totalidad. Entrando en una breve recapitulización de los mismos detallando específicamente su consecución a continuación:

- **El servicio deberá asignar un concepto dado de una terminología a un elemento de un modelo de datos.** Siendo éste el principal objetivo del proyecto, se consigue con éxito la vinculación de conceptos de las terminologias SNOMED CT, LOINC y HGNC con elementos del modelo de datos. Los datos sobre los que se ha probado la correcta funcionalidad del servicio son

los proporcionados por diversos socios de los proyecto INTEGRATE y EURECA. Aunque sería muy recomendable el poder tener acceso a una mayor cantidad de datos, específicamente de SNOMED CT, para poder ampliar las capacidades del servicio.

- **El servicio deberá proporcionar la integración de un conjunto de terminologías médicas** Las terminologías integradas específicamente en el actual estado del servicio son SNOMED CT, LOINC y HGNC. A pesar de que la terminología más difícil de integrar es SNOMED CT, y la es la de mayor complejidad, es la que realmente más se ha detallado y especificado el método elaborado. El resto de terminologías, LOINC y HGNC, también han sido integradas con éxito, a pesar de que su integración ha conllevado mucho menos esfuerzo que el realizado con SNOMED CT, y por ello ambas han sido integradas como partes anexas a la estructura de SNOMED CT. Comentar que en el estado actual del servicio otras terminologías pueden ser integradas con relativa facilidad. Ya que usando servicios como los proporcionados por BioPortal para traducir términos de los vocabularios ICD[22] o el de NCI[23] a SNOMED CT, se posibilita así su integración de una manera relativamente fácil.

- **Se deberá limitar a un modelo de datos ya existente basado en el HL7 RIM.** El modelo de datos al que se ha adaptado exitosamente el servicio realizado se encuentra en el anexo A.1. Éste modelo de datos se encuentra basado en el HL7 RIM se encuentra igualmente incluido en el anexo A.2. Tomando estos modelos como base se ha construido el servicio en función a ellos, el servicio logra integrar los datos de manera correcta en el actual modelo.

- **Integración con tecnologías ya empleadas** El servicio ha sido desarrollado desde un inicio en Java como un servicio web, por lo que logra integrarse con el resto de servicios que usan la misma tecnología, pudiendo ser desplegado en los mismos contenedores que el resto de servicios. Las tecnologías empleadas para el razonamiento semántico han sido las que actualmente estaban siendo usadas. Tan sólo se han realizado una serie de modificaciones en las ontologías de las terminologías, que son igualmente dispuestas en el repositorio Sesame, por lo que la creación del servicio no ha supuesto ningún cambio o requerimiento de tecnología adicional, salvo el uso adicional del dominio “provenance”.

- **Alto rendimiento del servicio** Como se han incluido en el capítulo de evaluación de la herramienta, los resultados finales de rendimiento cubren perfectamente las necesidades actuales, y posibles necesidades futuras del servicio.

La consecución de los objetivos que aquí se comentan como resueltos, han sido los pilares sobre los que se han basado las decisiones más básicas y fundamentales a la hora de desarrollar el servicio, sirviendo de guías durante toda su implementación.

Durante la realización del proyecto la mayor dificultad encontrada fue el estudio inicial de las terminologías y el modelo de datos. Durante dicho estudio se tomaron multitud de decisiones que tomaban como base las escasas directrices que podía encontrar en el proyecto TermInfo, pero que en la mayor parte de las situaciones no concluían nada específico o claro, debido a que el actual proyecto TermInfo 2 se encuentra aun siendo desarrollado, y la mayor parte de la información que es ofrecida en su wiki es incompleta o inconclusa. Sumado a esto, el escaso tiempo disponible para asimilar y asentar tal cantidad de información, conllevó que inicialmente se tomaran una serie de decisiones en el "mapping" de conceptos, que tras recibir nuevas fuentes de datos tuvieron que ser rectificadas tras comprobarse que eran erróneos o que inducían a falta de consistencia en la información integrada.

El proyecto TermInfo 2 también explica una serie de dificultades a la hora de integrar datos, ya que se producen solapamientos semánticos a la hora de integrar datos expresados en SNOMED CT en un modelo basado en el HL7 RIM. Estos solapamientos a pesar de ser mencionados no son resueltos, y han tenido que ser resueltos para poder integrar los datos presentes en el proyecto.

Otra de las dificultades encontradas durante la elaboración de éste proyecto es la falta de datos con los que poder trabajar y desarrollar el servicio. esto se debe a lo reticentes que son la mayor parte de las instituciones para facilitar datos médicos, aunque estos se encuentren totalmente anonimizados. Esto conlleva que al no tener una fuente abundo tener una fuente abundante de datos, y que los que se tengan sean administrados poco a poco, dificulte enormemente la labor del estudio previo para desarrollar el servicio. La mayor parte de los datos utilizados han sido facilitados del Instituto Jules Bordet, un centro belga dedicado al estudio y tratamiento del cáncer.

8.2. Líneas futuras

Aunque la consecución de los objetivos planteados es satisfactoria, existen una serie ideas pendientes que surgidas durante la elaboración del proyecto para mejorar el presente proyecto.

Los datos procedes de SNOMED CT integrados hasta el momento son los que

se están siendo gestionados actualmente en los proyectos INTEGRATE y EURECA. Ampliar los datos a integrar es una de las metas principales a conseguir con la continuación de éste proyecto, pero se trata de un tema que hay que hacer muy cuidadosamente, y que se vuelve cada vez mas complejo a medida que se aumenta la diversidad de datos SNOMED CT a integrar.

Otros de los aspectos a ampliar son las terminologías aceptadas por el servicio. Las terminologías que pueden ser más fáciles de incluir en el servicio podrían ser MedDRA[25], CDI[22] y el vocabulario de NCI[23]. Se puede considerar facilidad o proximidad en su integración en el estado actual del servicio, ya que con un mínimo estudio sobre estas terminologías se conoce de ciertos servicios a los que se pueden acceder para la traducción de conceptos expresados usados estas terminologías a conceptos SNOMED CT.

A pesar de que el servicio es totalmente autónomo, ha sido creado específicamente como parte del entramado de servicios para la carga y consulta de datos en el modelo de datos. La integración del TermBinding en los procesos ETL, en los que se da uso de las herramientas Mirth Connect y Pentaho - Kettle como se vió en el capítulo de tecnologías, es considerablemente uno de los puntos a mejorar.

La causa de esto son las escasas posibilidades de integración con la herramienta Kettle de Pentaho, ya que hasta el momento el proceso que se estaba aplicando era usar dicha herramienta para almacenar los datos de manera básica en un CDM básico, y una vez almacenados aplicarles el proceso de normalización para su total integración con el servicio TermBinding.

Por otro lado, la integración con la herramienta Mirth Connect es algo más plausible, pero debido a que es una herramienta que se está empezando a usar en el proyecto, la situación induce cierta reticencia si actualmente total adopción de la herramienta Mirth Connect para los procesos ETL, y la integración total del servicio TermBinding en dicho proceso, pero no como servicio web, si no como clase Java a la que acceder durante el proceso ETL.

8.3. Agradecimientos

Para finalizar, agradecer la oportunidad brindada por los proyectos europeos INTEGRATE y EURECA, para los que se encuentra englobado el presente proyecto fin de grado, donde he podido realizar una labor de investigación real aplicando todos los conocimientos adquiridos durante mi formación en la Facultad de Informática de la Universidad Politécnica de Madrid.

Además de poner en práctica los conocimientos adquiridos, he tenido que desarro-

llar e investigar muchos otros para poder desarrollar satisfactoriamente éste proyecto, complementando así mis estudios. Gracias a ésta oportunidad he podido formar parte de un equipo de trabajo y comprobar que hasta una de las pequeñas partes del proyecto global pueden suponer y ser de gran importancia, y que sin ellas, no se podrían conseguir los objetivos finales de estos proyectos, y poder haber desarrollado una de éstas partes es algo que realmente hace sentirte como que todo los años de formación sirven realmente de mucho. Yendo más allá, el formar parte de un proyecto en el que tiene como objetivo el ayudar a otros investigadores médicos en su labor para eliminar uno de las enfermedades con mayor índice de mortalidad en el mundo[24].

Bibliografía

[1] Perez-Rey D, Maojo V, Garcia-Remesal M, Alonso-Calvo R, Billhardt H, Martin-Sanchez F, Sousa A. ONTOFUSION: Ontology-based integration of genomic and clinical databases. Computers in Biology and Medi-cine Volume 36, Issues 7–8, July–August 2006; 712–730

[2] Alonso-Calvo R, Maojo V, Billhardt H, Martin-Sanchez F, Garcia-Remesal M, Perez-Rey D. An agent- and ontology-based system for integrating public gene, protein, and dis-ease databases. Journal of Biomedical Informatics, 2007 Feb; 40(1):17-29.

[3] M. Tsiknakis, D. Kafetzopoulos, G. Potamias, C. Marias, A. Analyti, and A. Manganas. Developing European bio-medical grid on cancer: The ACGT integrated project. Proc. HealthGrid 2006 Conf., Valencia, Spain, Stud Health Technol. Inf., vol. 120, pp. 247–58, Jun. 6–8.

[4] Oster S, Langella S, Hasting S, Ervin D, Madduri R, Phillips J, Kurc T, Siebenlist F, Covitz P, Shanbhag K, Foster I, Saltz J. caGrid 1.0: An Enterprise Grid Infrastructure for Biomedical Research. Journal of the American Medical Informatics Association 2008;15:138-149.

[5] Driving ex-cellence in integrative cancer research [Citada Abril 2013][Accedida Mayo 2013]. Disponible en: http://www.fp7-integrate.eu/index.php/project

[6] Enabling information re-Use by linking clinical REsearch and Care [Citada April 2013][Accedida Mayo 2013]. Disponible en: http://eurecaproject.eu/

[7] Informatics for Integrating Biology and the Bedside (i2b2) [Citada Abril 2013][Accedida Mayo 2013]. Disponible en: https://www.i2b2.org/software/index.html

[8] Observational Medical Outcomes Partnership (OMOP) [Citada Abril 2013][Accedida Mayo 2013]. Disponible en: http://omop.fnih.org/node/22

[9] Shagufta Umer, Muhammad Afzal, Maqbool Hussain, Khalid Latif, Hafiz Farooq Ahmad, "Autonomous mapping of HL7 RIM and relational database schema", Springer Science+Business Media, LLC 2011, 3 Junio 2011.

[10] HL7 International Organization [Citada Abril 2013][Accedida Mayo 2013]. Disponible en: http://www.hl7.org

[11] The Unified Modeling Language [Citada Abril 2013][Accedida Mayo 2013]. Disponible en: http://www.uml.org/

[12] Donnelly K, SNOMED-CT: The Advanced Terminology and Coding System for eHealth. Medical and care compunetics 3 2006: 279-290.

[13] McDonald CJ, Huff SM, Suico JG, Hill G, Leavelle D, Aller R, Forrey A, Mercer K, DeMoor G, Hook J, Williams W, Case J, Maloney P. LOINC, a Universal Standard for Identifying Laboratory Observations: A 5-Year Update. Clinical Chemistry. 2003; 49 No. 4: 624-633.

[14] Seal RL, Gordon SM, Lush MJ, Wright MW, Bruford EA. Genenames.org: The HGNC resources in 2011. Nucleic Acids Research. 2011; 39: 514-519.

[15] College of American Pathologists. SNOMED Clinical Terms Guide. Transforming Expressions to Normal Forms. Agosto 2006. http://www.cap.org/apps/docs/snomed/documents/transfor-mations-to-normal-forms.pdf

[16] Broekstra J, Kampman A, Harmelen F. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. The Semantic Web. 2002; 2342: 54-68.

[17] Web Service Glossary [Citada Mayo 2013][Accedida Mayo 2013]. Disponible en: http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/

[18] Web Services Description Language (WSDL) Version 2.0 [Citada Mayo 2013][Accedida Mayo 2013]. Disponible en: http://www.w3.org/TR/wsdl20/

[19] SOAP Version 1.2: Messaging Framework [Citada Mayo 2013][Accedida en Mayo 2013].Disponible en: http://www.w3.org/TR/soap12-part1/

[20] Jason Brittain, Ian F. Darwin. Tomcat: The Definitive Guide. Junio 2003.

[21] IEEE Recommended Practice for Software Requirements Specification. IEEE Standard, 1998.

[22] International Classification of Diseases (ICD) [Citada Mayo 2013][Accedida Mayo 2013] Disponible en: http://www.who.int/classifications/icd/en/

[23] National Cancer Institute [Citada Mayo 2013][Accedida Mayo 2013] Disponible en: http://www.cancer.gov/cancertopics/cancerlibrary/terminologyresources

[24] Leading causes of death by broad income group (2008) [Citada Mayo 2013][Accedida Mayo 2013] Disponible en: http://who.int/mediacentre/factsheets/fs310/en/

[25] Medical Dictionary for Regulatory Activities [Citada Mayo 2013][Accedida Mayo 2013] Disponible en: http://www.meddramsso.com/

[26] Cheetham E, H. Dolin R, Markwell D, Curry J, Gabriel D, Hausam R, Knight B, Rector A, Spackman K, Townend I. Using SNOMED CT in HL7 v3 Implementation Guide, Release 1.5. 2008.

26

Anexos

Anexo A

Esquema de los Modelos de Datos

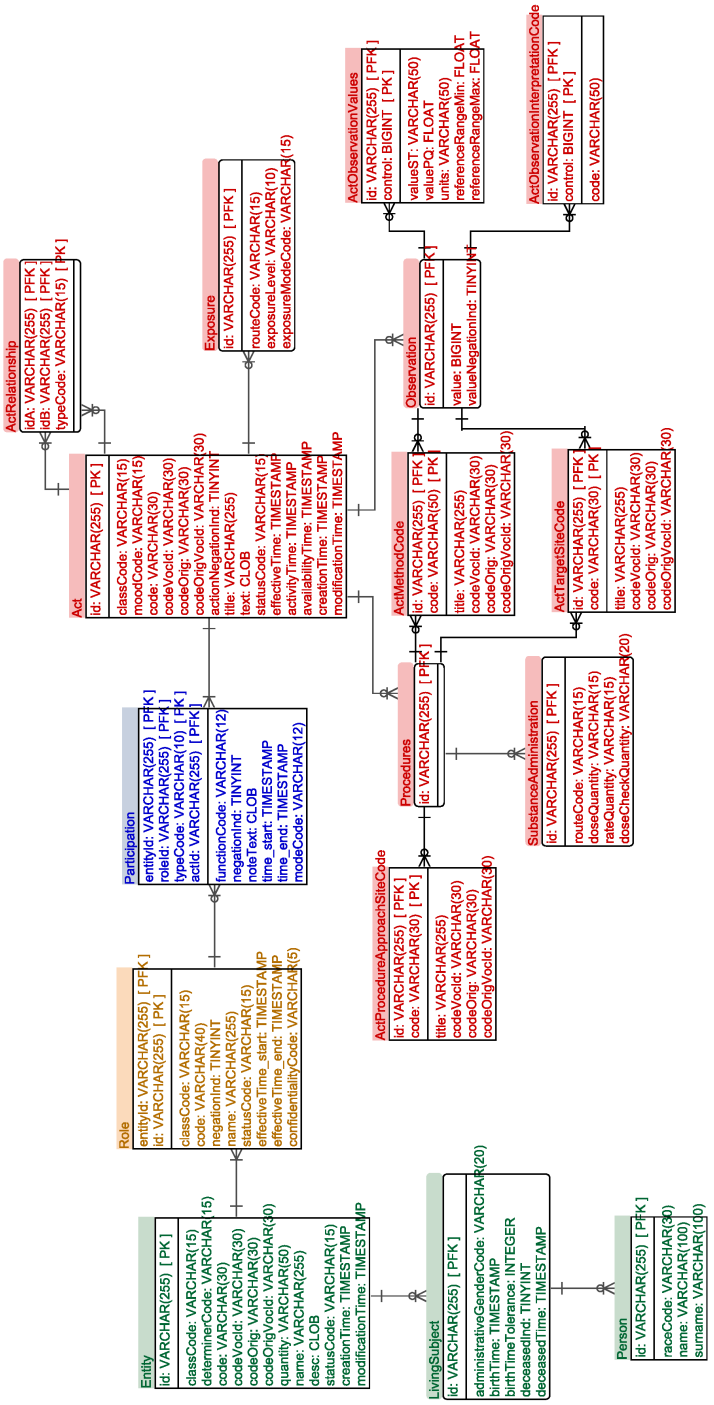

Figura A.1: Diagrama del Modelo Común de Datos

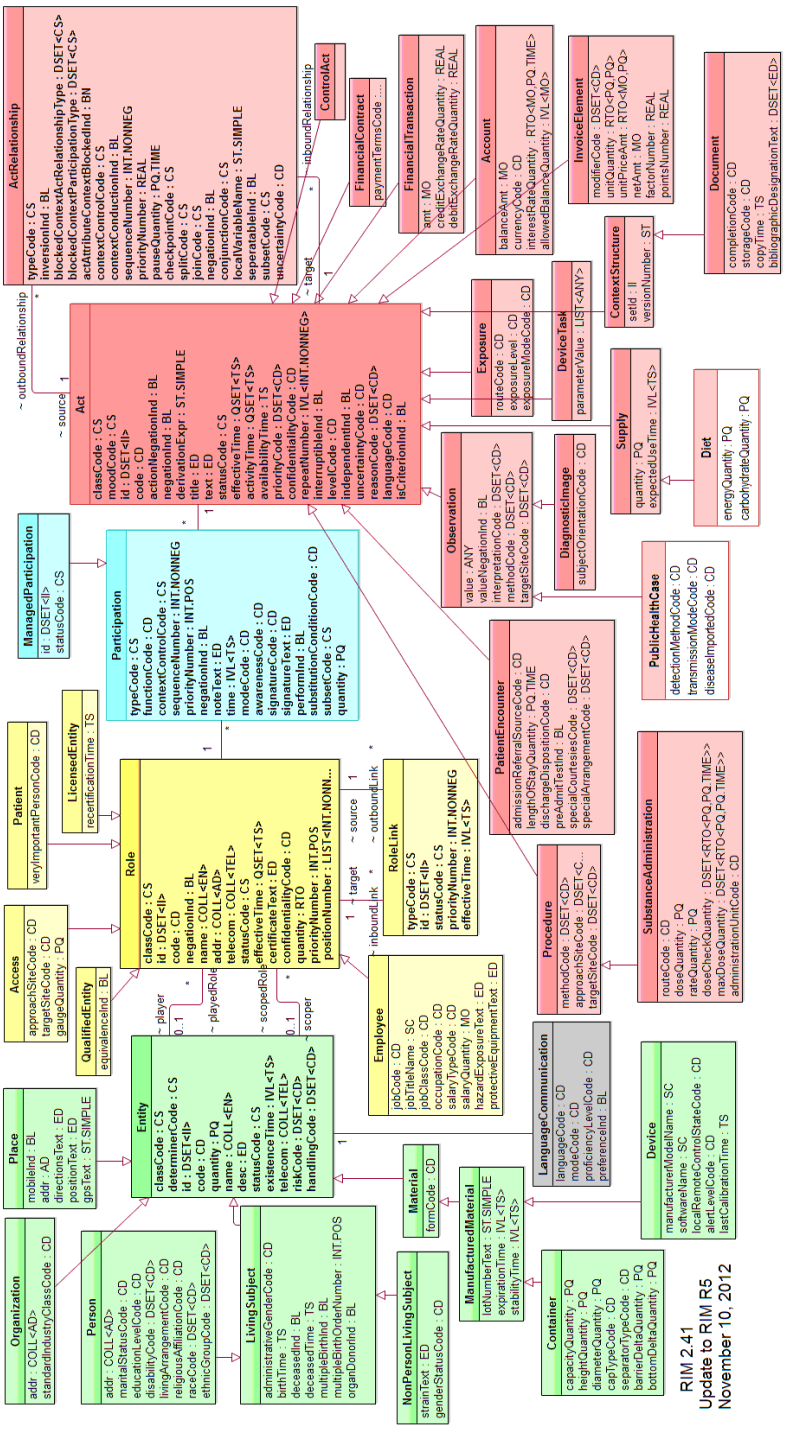

Figura A.2: Esquema del modelo HL7 RIM

Anexo B

Artículos científicos publicados

Analyzing SNOMED CT and HL7 Terminology Binding for Semantic Interoperability on Post-Genomic Clinical Trials

Santiago Aso[a], David Perez-Rey[a], Raul Alonso-Calvo[a], Antonio Rico-Diez[a], Anca Bucur[b], Brecht Claerhout[c], Victor Maojo[a]

[a] *Biomedical Informatics Group, DIA & DLSIIS, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo S/N, 28660 Boadilla del Monte, Madrid, Spain.*
[b] *PHILIPS Research Europe, High Tech Campus 34 Eindhoven, Netherlands*
[c] *Custodix NV, Kortrijksesteenweg 214b3, Sint-Martens-Latem, Belgium*

Abstract

Current post-genomic clinical trials in cancer involve the collaboration of several institutions. Multi-centric retrospective analysis requires advanced methods to ensure semantic interoperability. In this scenario, the objective of the EU funded INTEGRATE project, is to provide an infrastructure to share knowledge and data in post-genomic breast cancer clinical trials. This paper presents the process carried out in this project, to bind domain terminologies in the area, such as SNOMED CT, with the HL7 v3 Reference Information Model (RIM). The proposed terminology binding follow the HL7 recommendations, but should also consider important issues such as overlapping concepts and domain terminology coverage. Although there are limitations due to the large heterogeneity of the data in the area, the proposed process has been successfully applied within the context of the INTEGRATE project. An improvement in semantic interoperability of patient data from modern breast cancer clinical trials, aims to enhance the clinical practice in oncology.

Keywords:

Semantic Interoperability, Clinical Trials, SNOMED, HL7 RIM, INTEGRATE project, Breast Neoplasm, Terminology Binding

Methods

The SNOMED CT Normal Form allows to homogeneously access the data. It also simplifies the storage and retrieval of pre and post-coordinated expressions as well as the implementation of the HL7 recommendations. The HL7 TermInfo Project initiative provides several guidelines to achieve a binding of the normalized concepts into the chosen model (HL7 Reference Information Model), but it also presents some open issues to solve like overlaps. After solving those issues for a concrete type of data (patient data of breast cancer clinical trials), the original terminologies are enriched manually with information extracted from the binding analyzed throughout the study of the HL7 TermInfo Project. And they are finally stored using an ontology representation language (Ontology Web Language - OWL). This information is afterwards used to perform queries to the integrated repository.

Results

Clinical data has been represented with the proposed approach of a HL7 RIM based Common Data Model (CDM) and a core dataset expressed with terminologies SNOMED CT and LOINC. HL7 v2 and v3 messages were loaded into the Common Data Model with an Extract, Transform and Load tool. The data comes from clinical trials conducted over 50 patients and a total of more than 2,400 acts performed during these trials. The distribution of the data among classes of the Common Data Model, resulting from applying the previous methods is shown on Figure 1.

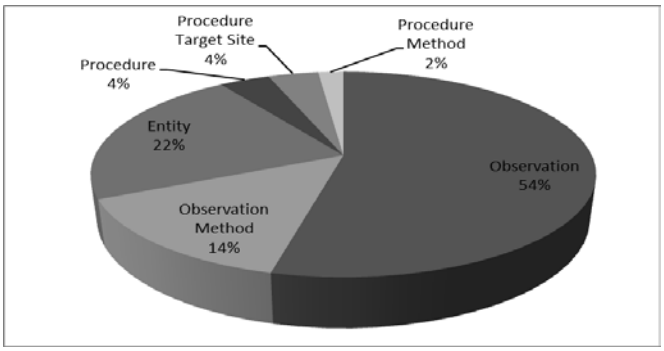

Figure 1 Distribution of data available at the INTEGRATE project.

Conclusions

We have described the process proposed to provide a terminology binding between SNOMED CT and LOINC with a CDM based on the HL7 RIM. This solution has been tested within the INTEGRATE European project to provide semantic interoperability for multi-centric breast cancer clinical trials. The terminology binding can provide semantic interoperability to ensure the sustainability of modern clinical trials in oncology. That is an essential requirement to gather complex, multiscale and highly heterogeneous data from post-genomic cancer patients.

References

[1] Paraiso S, Perez D, Alonso R, Claerhout B, Schepper K, Hennebert P, Lhaut J, Leeuwen J, Bucur A. Semantic interoperability solution for multicentric breast cancer trials at the INTEGRATE EU Project. In Proceedings of the 6th International conference in Health Informatics 2013.

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Fri Feb 14 20:10:55 CET 2014
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)